ACORN USER

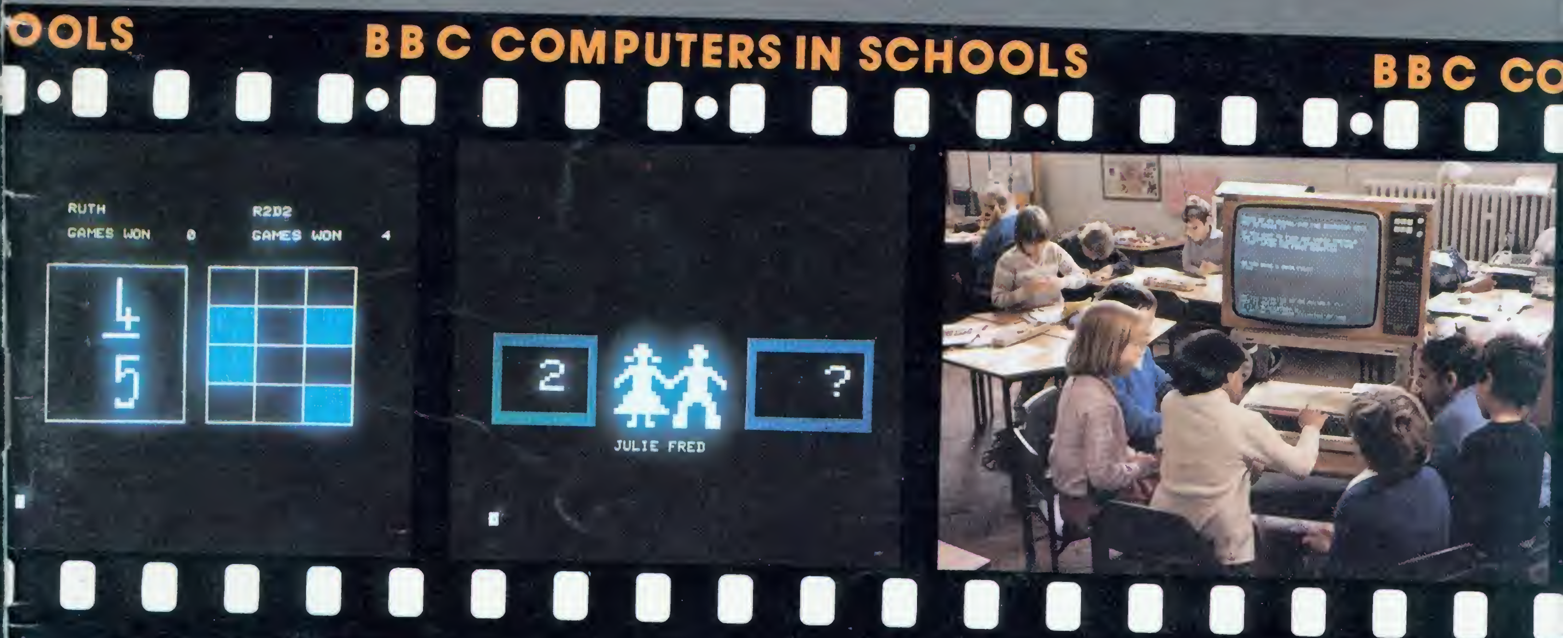
For BBC micro, Atom and Econet users

January 1983 £1

Dealing with discs

Pet printing

BBC TV series



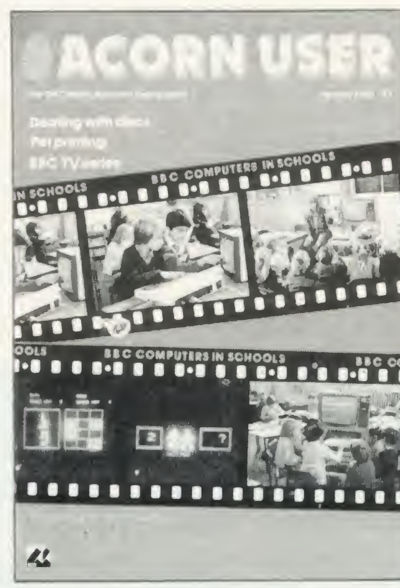
First in a 12 part series

CONTENTS

ACORN USER

JANUARY 1983, NUMBER SIX

Editor
Tony Quinn
Managing Editor
Jane Fransella
Sub Editor
Ann Nimmo
Production
Susie Home
Marketing Manager
Paul Thompson
Promotion Manager
Pat Bitton
Editorial Secretary
Jane Lake
Publisher
Stanley Malcolm
Typesetters and Designers
GMGraphics, Harrow Hill
Graphic Designer
Phil Kanssen
Printed in Great Britain
by E.T.Heron & Co. Ltd
Advertising Agents
Dealer Deals Ltd.
20 Orange Street
London WC2H 7ED
01-930 1612
Distributed to the News Trade
by Magnum Distribution Ltd.
72-8 Fleet Street,
London EC4Y 1HY.
Tel: 01-583 0961
Telex: 893340 Magnum G.
Published by
Addison-Wesley Publishers Ltd.
53 Bedford Square,
London WC1B 3DZ
Telephone: 01-631 1636
Telex: 881 1948
ISSN: 201-17002 7



Cover design by Chris Gilbert
Photography by Malcolm Aird

How to submit articles

You are welcome to submit articles to the Editor of *Acorn User* for publication. *Acorn User* cannot undertake to return them unless a stamped addressed envelope is enclosed. Articles should be typed or computer written. Black and white photographs or transparencies are also appreciated. If submitting programs please send a cassette or disc. Payment is £50 per page or pro rata. Please indicate if you have submitted your article elsewhere. Send articles, reviews and information to: The Editor, *Acorn User*, 53 Bedford Square, London WC1B 3DZ.

Coming soon in *Acorn User*:

● Dynamic procedures ● Software reviews ● Science programs ● Hardware modifications for the Beeb ● Atom 3D graphics ● Word processing reviews ● Guide to printers ● More on Atom utility boards ● Games listings ● Sorting out machine code errors ● BBC telesoftware launch ● Computer Literacy Project – assessing the first year ●

Subscription Information

For UK Subscriptions, send your cheque or postal order made payable to Addison-Wesley Publishers Ltd to:
MAGSUB (Subscription Services) Ltd, Ground Floor Post Room
Oakfield House
Perrymount Road
Haywards Heath
West Sussex RH16 3DH

Annual subscription rates

UK	*£15*
Europe	*£18*
Middle East	*£20*
The Americas and Africa	*£22*
Rest of the World	*£24*

WILL ACORN COME GOOD?

CAN Acorn Computers turn over a new leaf and shake off its bad reputation for delivery and marketing – a situation compounded by poor support, unsettled dealers and permanently engaged phone lines? This is the question many users and supporters of the BBC micro are asking as a New Year begins.

The reasons behind the problems are many – impossible promises, the unusual nature of the BBC–Acorn link, and the way sales have outstripped the company's capacity to support them.

But now is the time for consolidation, and pressure put on Acorn seems to be bearing fruit. The company's headquarters in Cambridge – a converted waterworks – has been expanded, Vector Marketing has at last got to grips with distribution, and more staff are being taken on to improve customer, dealer and sales support.

And the name of the game as 1983 unfolds will be system development to fulfil the promise of expandability given by the BBC machine's design. Expansion is what the Beeb is all about – the ability to use networks, telesoftware, the Tube, second processors and language ROMs.

Then we have the Electron, its launch now set for March. Acorn is playing it cool on this front and seems determined to avoid the production and delivery disasters which have hit the company's image with the BBC micro.

Will Acorn come good? Will customers accept Acorn's statement on operating systems? Only time will tell.

For the final word in this editorial, I hand you over to Kate Burns, aged 10. Her poem sets out to explain what this computer madness is all about.

Clever little bits of nothing
Telling this, telling that.
And when you start there's no ending,
It gets in heart, mind and brain.
It's like the plague
Catching, spreading
Moving fast,
People dreading that they'll catch
COMPUTER PLAGUE

Second processor trials going well

'New 32-bit Gluon is something to look forward to'

PROTOTYPES of Acorn's 32-bit second processor, the Gluon are 'working fine'.

The Gluon is built around three 32-bit chips and is designed to increase the processing speed and memory capacity of the BBC micro.

This is achieved by connecting the Beeb and second processor through the Tube, which is Acorn's high speed data transfer interface.

The chips used are National Semiconductor 16032s. 'These are the only true 32-bit microprocessors available,' said Hermann Hauser, Acorn's managing director.

'We are very pleased with the performance of the 32-bit chips, the prototypes are working fine. This is something to look forward to,' he added.

With the Gluon, the BBC micro will be able to match the processing power of some present day mini-computers, said Acorn, at a fraction of the cost.

This makes high-level languages available such as Fortran and Cobol, allows up to 96k of RAM to be used and puts the BBC machine at the top of the benchmark stakes.

A review of last year's *Personal Computer World* benchmarks put the Beeb in the third place, knocking spots off Apples, Sinclairs, Dragons etc. With a second processor the Beeb comes out number one, says Hauser.

Two other second processors, a Z80 board and 6502 (the same 8-bit chip as the Beeb uses), are well into the development stage and even these transform the timings.

Once these second processors are available, the BBC machine will take on a serious business role, with the ability to run CP/M software – the standard business language.

This will enable Acorn to develop its own business machine and step into the market against the likes of Torch. The existence of the Torch business machine is particularly galling, as this uses BBC boards and has captured a market Acorn has been slow to move into.

Deliveries and discs

UPGRADED BBC micros with the disc interface are now being sent out with a note to tell users how to carry on using cassettes.

Many customers did not know the *TAPE command was needed to switch from the disc to cassette filing modes.

Joe Telford, our hints and tips writer, had the same problems. He explains how he sorted the disc system out on page 19.

IF YOU have ordered a BBC micro by courier from Vector Marketing, it should arrive within 10 days, says Acorn.

The company hopes it will take less than this, and credit card orders should be processed faster than cheques.

Some computer industry pundits have pointed out that bad publicity may have convinced the public that BBC machines are not available, just when they are in fact being stockpiled by Acorn.

Five-station networks

IF YOUR system isn't big enough for Econet, try Bootnet. This software to control a network of four BBC machines with a Research Machines 380Z as file server costs less than £35.

And modifications to replace the 380Z with a BBC micro should have been completed by now.

Bootnet has been on trial for three months at 18 Birmingham schools. It was designed as a cheap introduction to networking and to provide disc storage for several micros.

The 380Z can be used while acting as a file server, except the four station micros have priority – which at times of high demand means the 380Z user will just have to wait.

No special hardware is required, as the system runs through the I/O (input/output) port of the computers. A five-way cable connects the machines, with no screening as this is thought to be unnecessary over short distances (10m of cable are provided in the largest standard package).

Bootnet software for the BBC micro will be provided on cassette.

Details from Mace, West Midlands Regional Centre, Four Dwellings School, Dwellings Lane, Quinton, Birmingham B32 1RJ.

Customer support

ACORN has re-affirmed its commitment to customer support, particularly in the education sector.

Joint managing director Hermann Hauser has confirmed that the educational series division within the company will be doubled in size – and recruitment has already begun.

Complaints by customers about the lack of support and fears among education advisors that back-up might not be good enough have spurred these efforts.

The dealer and sales support sections within Acorn are being revamped and expanded to meet increased demand.

Orders have already been placed by primary schools for computers, and early figures suggest BBC machines have captured over half of these.

The total number of primary schools is over 25,000, all of which are eligible to buy micros at half price with Department of Industry support.

The offer to secondary schools has now finished, and the Government seems satisfied that its idea has worked, with virtually all schools having purchased at least one micro.

Junior education minister William Shelton announced this at the recent MEP Microprimer launch and added that it was something everybody could be proud of.

Training film

A VIDEO film is available from local MEP offices to demonstrate teachers training and the use of micros in schools.

It is called Micros in Primaries – Starting Out and was shot in three schools, and on a weekend training course.

The 40-minute video underlines three aspects of computer education: practice and reinforcement; simulation and data handling.

Producer Robert Veale made the film for the DES, and hopes to start on a follow-up later in the year.

MEP launches major package for training teachers

Just one of the books in MEP's Microprimer

TRAINING 50,000 primary teachers to use micros is no small task – and the organisation set up to do just this has produced no small training package.

The Microelectronics Education Programme (MEP) has had 500 children checking the 30 programs written by 110 people for its Microprimer package. Another 20 programs are under development.

Microprimer will be presented to teachers with their micros on the training course which forms part of the Government's scheme.

Each pack is specific to one of the three sponsored machines and consists of: introduction, reader, study text, activity guide, four case studies on cassette, fact file (database programs), and 11 pieces of software.

The remaining software will follow in three other packs, and two more are planned. Most of the programs were specially commissioned and the first four packs are free. The next two will be available at a subsidised price (around £15 per pack, including explanatory booklet).

The material is well packaged and designed, and most of it comes in coffee-proof folders.

Programs cover spatial perception, maths, quizzes and language in pack one, as well as a monitor testcard and program to check the cassette player volume controls.

Schools have written these programs with financial support from the MEP – and the organisation is keen to stress the need for schools to come forward with more ideas.

Beeb trade exhibition

THE first major BBC microcomputer system trade exhibition will be opened on January 5. Entrance will be by invitation only at the National Microelectronics and Electronics Centre of London's World Trade Centre.

Unfortunately, it will not be open to the public, but to dealers and those who have played a part in developing the BBC's Computer Literacy Project.

The three-day exhibition features a wide range of materials and services associated with the project, including TV programmes from the new series (see page 8) and the *Computer Programme*, a display of telesoftware, new hardware and software, the NEC correspondence course, and information from Broadcasting Support Services.

Turtle writer takes the floor

TURTLES – for use with the *Logo* language – are now being made available for the BBC micro.

These devices can be described simply as pens on wheels, which are housed in a plastic dome. The turtle obeys commands given by the computer and draws on paper laid out on the floor what is usually seen on the TV display.

For £350 buyers receive the Turtle, an RS232 interface, connecting cable, and software.

The Turtle is based upon a version designed at Edinburgh University, and comes with a one-year guarantee.

Jessop Electronics, who make the device, say they are adapting it to run on all the versions of *Logo* they can find. The control program is usually in EPROM housed in the RS232 interface, which runs at 1200 baud.

The 1.0 BBC operating system is needed, and Jessop Electronics can be contacted at Unit 5, 7 Long St, London EC2 8HN

Printer bug in Atom BBC board

A BUG has been discovered in the BBC Basic conversion board for the Atom, which is produced by Acornsoft.

The problem affects Centronics-type printers such as the Seikosha, and means they will only print with double line spacing when used in BBC mode.

This is because the printer expects the linefeed to be suppressed, which does not happen with the conversion board's operating system ROM. (The BBC micro suppresses the linefeed by default.)

Acornsoft had not realised the problem until it was pointed out by an *Acorn User* reviewer who is putting the board through its paces

The company has promised to ensure the bug is ironed out for future issues of the ROM. Anybody who is having problems with the board should contact Acornsoft direct at 4a Market Hill, Cambridge CB2 3NJ.

Electronic treasure trove

ELECTRONIC treasure seekers will soon have another target – this time from Acornsoft and worth a total of £2,000.

Castle of Riddles is the name of the third adventure game from Acorn's software arm, but the first to offer a real reward.

Players must solve a series of problems to win, and answer a number of riddles.

David Johnson-Davies, managing director of Acornsoft is giving few clues away, apart from the fact that the final riddle is encrypted in a code.

To prevent anyone having an advantage, retail and mail order copies will be released at the same time.

Adventure games are a fast-expanding part of the software business and several companies are offering rewards to the first person to crack the game.

Castle of Riddles comes on cassette, and a disc version is planned. Details of the competition are included with every copy of the game.

Dial a golden fortune

READERS who want to go after the £6000 Golden Sundial of Pi had better get their skates on – Sinclair users have had a three month start.

The Sundial is being offered to the first person who finds the electronic substitute in the adventure game *Pimania*.

However, the 32k BBC version should only now be coming available, while Sinclair users have been buying the game since October.

Christian Penfold of Automata, makers of *Pimania*, explained that the BBC translation had been delayed, and might not be available by Christmas.

The Sundial was commissioned by Automata and is made from gold and precious stones – and it can tell the time!

BBC series hits US

IT SEEMS the BBC has beaten Acorn into the US. Sales of its TV series *The Computer Programme* now stand at more than 60 to the Americans.

The series, each of ten episodes, are destined for schools and have been sold by the BBC's distributor Films Incorporated.

Acorn in the meanwhile has been quiet about its transatlantic activities since the news in November that a subsidiary has been set up to market the BBC micro over there.

Teachers' soft

A GROUP of teachers has set up Chalksoft to produce primary and middle school software for the BBC model B. Five programs are available – *Metrics* (9.95), *Invisible Man* (£5.95), *Sequences* (£5.95), *Angle* (£6.95) and *Inkosi* (£5.95). Three others are being developed, and all make use of colour and sound.

Details from Chalksoft, Lowmoor Cottage, Tonedale, Wellington, Somerset, TA21 OAL.

OU to launch computing course

THE Open University is about to embark on its own programme to teach teachers about computing and the BBC micro.

Spring sees the launch of an 'awareness pack', which will sell for about £20. It follows the usual OU methods of distance learning and written material is provided with cassette software to illustrate the ideas.

The pack is pitched at an elementary level and two follow-up courses are already planned.

The first, scheduled for later this year, covers educational software, the second, microelectronics and hardware (1984). The hardware pack will include a simulation board. Videotapes should be available to illustrate both later courses, and a TV series is under discussion.

contact Helen Boyce, Micros in schools project, Maths faculty, Open University, Walton Hall, Milton Keynes, Bucks.

Joe Telford adds an aside to his hints

First, some limbering up. Ready? Type VDU7 and press RETURN. OK, just testing control character 7, which produces a rather pleasant 'beep'. Pressing CTRL–G should do the same. Readers might be wondering what CTRL–G means. It is really very much in the same mould as the use of SHIFT–5 to produce a % sign. Let's look at the key sequence for CTRL–G: press down on CTRL key and hold it; tap G; release G; release CTRL key.

I mention these finger gymnastics because the spectacle of 30 adults at a recent meeting failing to make key sequences like CTRL–G and SHIFT–BREAK shook me somewhat.

However, back to the Beeb beep. There are a number of instances where a user wishes to produce sounds without the hassle of setting up the Sound command. To this end I've been playing with the *FX commands, and have found that some of them affect the VDU7 facility. *FX210,1 turns the beep off, *FX210,0 turns it back on.

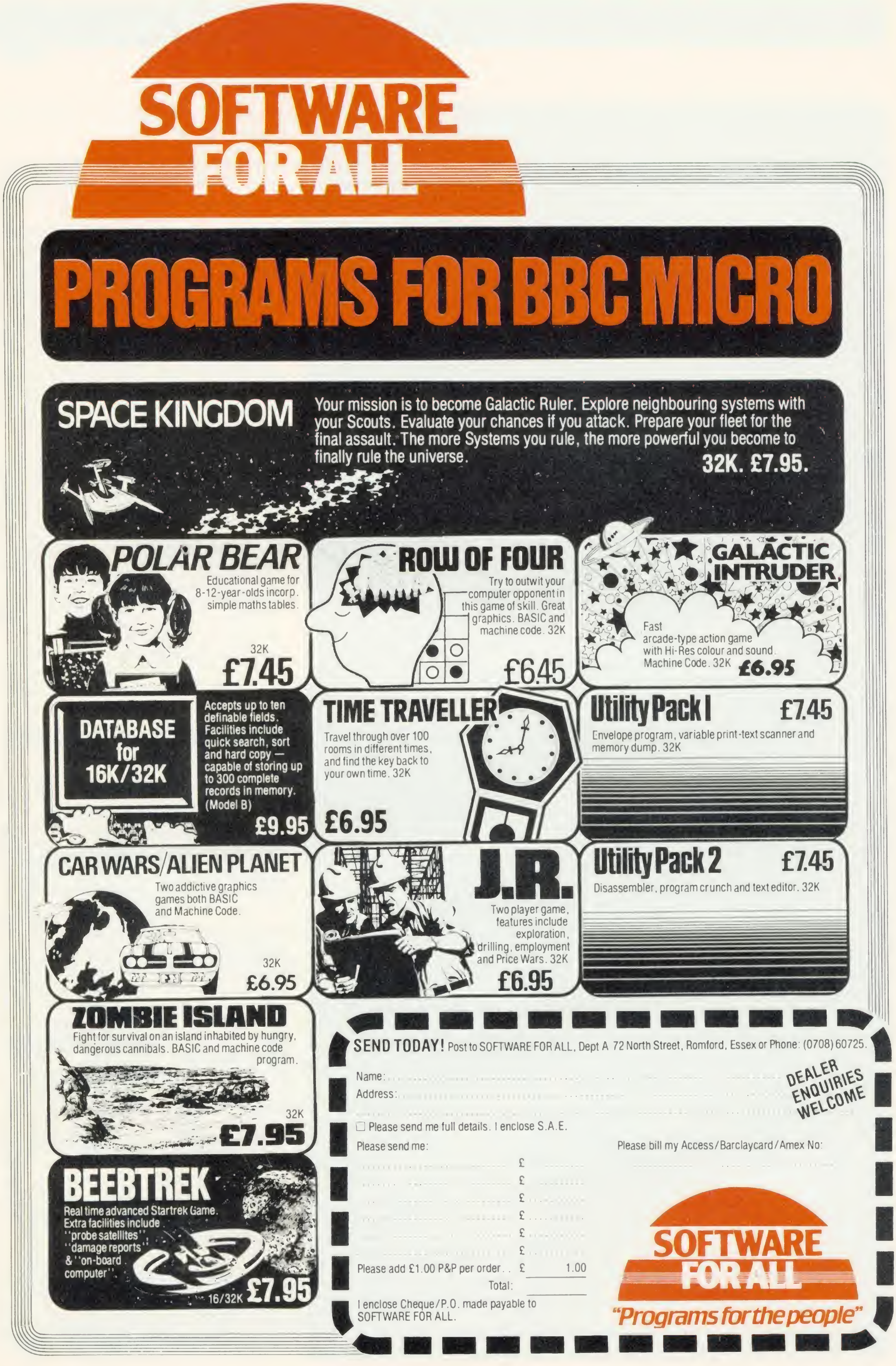
We can also control the pitch of the beep. *FX213,p does this where p is a number from 0 to 255. 0 is the lowest note and 255 is the highest. The default value is 101 (as near as I can hear). Not only can we control the pitch, but also the duration is changeable via the *FX commands. This can be done with *FX214,d. Here, 'd' is the duration of the VDU7 beep, in 20ths of seconds. The normal beep duration is set by *FX214,7. Another control parameter for VDU7 is the voice used by the beep. *FX211 changes the voice from 1, the default voice, to 2, 3, or 0 (for white noise).

I believe these *FX commands only operate from machine operating system 1.00, though they should exist on all later OS versions (to find out which you have, type *FX0). A couple of useful 'beeps' can be set up with these *FX commands.

Steam train beep.
Type *FX211,0
then CTRL–G

BBC time pips.
Type
*FX211,1
*FX213,149
*FX214,5
then CTRL–G

To conclude this section I am lead to believe that *FX212 controls the volume/envelope number of the VDU7 command. I cannot confirm this, as no information was available from Acorn about any of the above commands other than *FX210. However, I used *FX212 to produce strange effects when incorporated with an envelope which had been set up. A genuine puzzle for readers!


SOFTWARE FOR ALL
PROGRAMS FOR BBC MICRO
SPACE KINGDOM
Your mission is to become Galactic Ruler. Explore neighbouring systems with your Scouts. Evaluate your chances if you attack. Prepare your fleet for the final assault. The more Systems you rule, the more powerful you become to finally rule the universe.
32K. £7.95.
POLAR BEAR
Educational game for 8-12-year-olds incorp. simple maths tables.
32K
£7.45
ROW OF FOUR
Try to outwit your computer opponent in this game of skill. Great graphics. BASIC and machine code. 32K
£6.45
GALACTIC INTRUDER
Fast arcade-type action game with Hi-Res colour and sound. Machine Code. 32K
£6.95
DATABASE for 16K/32K
Accepts up to ten definable fields. Facilities include quick search, sort and hard copy — capable of storing up to 300 complete records in memory. (Model B)
£9.95
TIME TRAVELLER
Travel through over 100 rooms in different times, and find the key back to your own time. 32K
£6.95
Utility Pack I
£7.45
Envelope program, variable print-text scanner and memory dump. 32K
CAR WARS/ALIEN PLANET
Two addictive graphics games both BASIC and Machine Code.
32K
£6.95
J.R.
Two player game, features include exploration, drilling, employment and Price Wars. 32K
£6.95
Utility Pack 2
£7.45
Disassembler, program crunch and text editor. 32K
ZOMBIE ISLAND
Fight for survival on an island inhabited by hungry, dangerous cannibals. BASIC and machine code program.
32K
£7.95
BEEBTREK
Real time advanced Startrek Game. Extra facilities include "probe satellites" "damage reports" & "on-board computer".
16/32K £7.95
SEND TODAY! Post to SOFTWARE FOR ALL, Dept A 72 North Street, Romford, Essex or Phone: (0708) 60725.
DEALER ENQUIRIES WELCOME
Name:
Address:
☐ Please send me full details. I enclose S.A.E.
Please send me:
Please bill my Access/Barclaycard/Amex No:
£
£
£
£
£
£
Please add £1.00 P&P per order £ 1.00
Total:
I enclose Cheque/P.O. made payable to SOFTWARE FOR ALL.
SOFTWARE FOR ALL
"Programs for the people"

The new TV series will be shown on Monday nights from January 10 to March 14 on BBC1. Repeats will go out on Sunday at 12.35 from January 16. Transmissions for schools will also be shown on Mondays at 15.05, but on BBC2

THE ten programmes in this new series will be:

The versatile machine looks at a basic microcomputer system and the way it can be expanded to do different things – such as controlling a piece of machinery, word processing, communicating down the telephone line, plotting a drawing, and so on, as well as playing games and doing calculations. This programme looks inside the machine to see a little of how it works. The idea is to demystify some of the terminology of microcomputing which owners of machines are likely to meet, and to set the scene for later programmes.

Getting down to Basics is the first of two programmes which look at writing a simple program. There are only a few fundamental programming structures which apply even in large programs, and in all languages and on all machines. Ian McNaught Davies writes a simple program and progressively enlarges it. Also, he looks at how first time users have reacted to having a machine and the use they have made of the software available, and at how to judge good software as a consumer. (See December issue).

Strings and things opens by looking at the way the computer can handle words. We then look at the last major programming principle – the use of the subroutine or procedure – (reinforcing the principles seen earlier). The series now moves on to look at managing a large program.

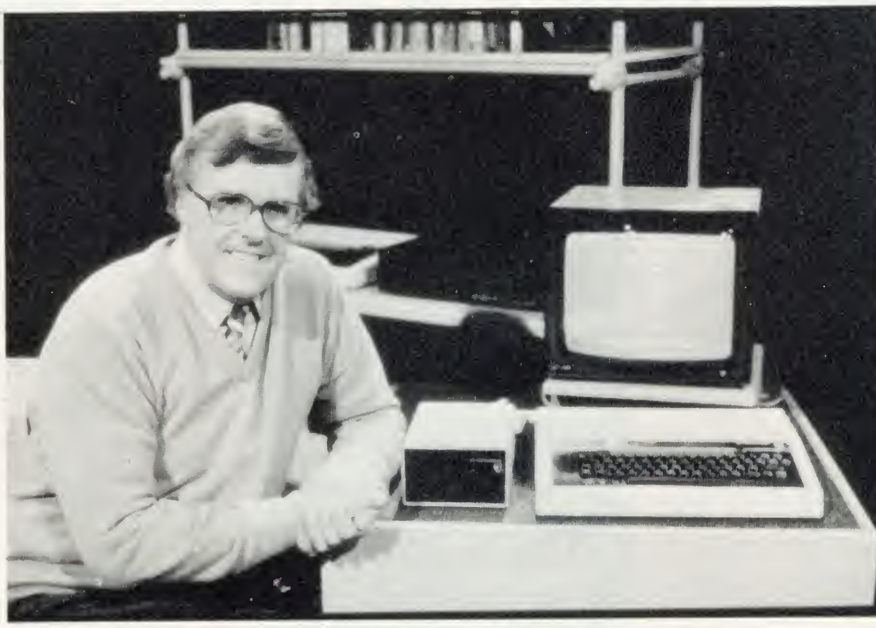
Graphics are handled in many ways, as each machine has different detailed ways of producing lines, colour and movement on the screen. However, many of the fundamental principles are common and even the humblest machines can produce dramatic effects, mimicking professional systems.

Data bases are one of the fundamental techniques and one of the most important and useful things the micro can handle. This programme takes on some of the simple ideas in data processing – a subject which fills many people with bewilderment.

In business how do you go about using a microcomputer? What can it do for you – and what are the pitfalls? Where can you go for advice that's likely to be objective? What are the most popular pieces

of software used in business? Just some of the questions raised, and hopefully, answered.

Getting away from Basic examines the normal programming language of the personal micro-computer which uses a vocabulary of commands built up from ordinary English. But for many purposes, friendlier languages are available which make writing aprogram easier – and this is particularly true of programs which appear to show that the computer is intelligent – for example in programs which 'learn'.



The micro in control considers one of the most important potential uses of the computer; monitoring and controlling things.

Programme nine is based around non-keyboard input and non-screen output. It is not always necessary to use the keyboard to put information into the computer and not always necessary to use a screen for output. This programme examines some of the fascinations of computer aided design, speech creation and music on the micro.

Communication using the micro is the final subject tacked in the series – using the telephone to make contact with the large public and private databases or to communicate with another individual, and at how it is now possible to use the broadcast television signal to receive telesoftware. Then there is networking – where a number of micros in a single building or area communicate with each other and make use of common facilities like disc storage and expensive printers.

EDUQUEST

THE PROFESSIONAL SOFTWARE PEOPLE

EDUQUEST

multiple choice questions and answers pack for BBC model B

WCC

Multiple Choice Question & Answer Pack

- *Specially designed for educational users*
- *For use with the BBC Model 'B' Micro computer*
- *£25.00 including VAT and P & P*

Pack Contains

- *Instruction manual*
- *Master input mode*
- *3 Reception modules, each display questions and answers in a format suitable for students of different abilities and age groups*
- *Blank tape for data.*

THE PACKAGE CAN BE USED BY STUDENTS FROM AGE 6 UPWARDS BY SELECTING THE MOST SUITABLE RECEPTION MODULE, IT MAY ALSO BE USED IN BUSINESS FOR APTITUDE TESTING AND IN THE HOME FOR EDUCATIONAL QUIZZES.

Word Processing Pack

- *A simple to use tape-based word processing package*
- *Ideal for the small business or home user*
- *For use with the BBC Model 'B' Micro computer*
- *£10.00 including VAT and P & P.*

EDUQUEST

Word Processing Pack

WCC

EDUQUEST

WCC

Minefield

WCC

Minefield

- *An entertaining family game*
- *Try and cross the minefield without blowing yourself up!*
- *3D graphics on Model 'A' and 'B'*
- *Only £5.95 including VAT and P & P.*

TO: **EDUQUEST** *1 Thames Avenue Windsor Berkshire SL4 1QP Tel: Windsor (07535) 58079*

Please send me:
(Tick as required) Multiple Choice Pack ☐
Word Processing Pack ☐
Minefield Model 'A' ☐
Model 'B' ☐

I enclose cheque for £..................or charge my Access/Visa/Trustcard Account No:

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Access
Buy It with Access
VISA
ACCESS AND BARCLAYCARD/VISA WELCOME

Signature..
Name...
Address..
....
....
....

TWO TO PASS

In the third of their articles on assembly language, John Ferguson and Tony Shaw explain why two passes may be needed to translate Basic into machine code.

The program examples encountered in the first two parts of this series have illustrated some of the special features of the BBC machine. Some of these features, such as passing information to machine code routines through A%, X% and Y%, are not only useful but, like the assembler itself, easy to use. The time has come to take a look at when and how two 'passes' of the assembler are used in translating programs to machine code.

As we have seen, the BBC assembler allows us to define labels at the start of a Basic program eg:

```
SCREEN = &3C00
```

or as a program address within the assembler source program, eg:

```
25.START LDA#&41
```

However, problems arise when a label is referenced before it is defined.

```
10 P% = &1500
20 [
30.START          JMP FINI
40                LDA #&21
50.FINI           RTS
60 ]
```

The label FINI is encountered in the program before it is defined. If we try and assemble this program the assembler assigns the value &1500 to the lable START since this is the start of the program. JMP will then be translated into its corresponding machine code value (4C) but when it attempts to put a value to FINI (where it has to jump to) it has no idea – since the value of FINI is not defined until later in the program.

The end result is the assembly stops with the Beeb displaying an error message:

'No such variable at line 30.'

To overcome this requires two modifications to our program. The first involves using the OPT statement (*User Guide* page 314).

This statement normally follows the square bracket at the start of the assembler program and is followed by a number that determines its action.

OPT 0 assembler errors suppressed, no listing
OPT 1 assembler errors suppressed, listing
OPT 2 assembler errors reported, no listing
OPT 3 assembler errors reported, listing

Adding OPT 1 to our program will take us a stage further. The assembler will now pass through the entire program since error messages have been suppressed.

```
10 P% = &1500
20 [OPT 1
30 . START          JMP FINI
40                  LDA #&21
50 . FINI           RTS
60 ]
```

listing

```
1500    OPT 1
1500 4C 00 15 . START JMP FINI
1503 A9 21              LDA#&21
1505 60                 .FINI RTS
```

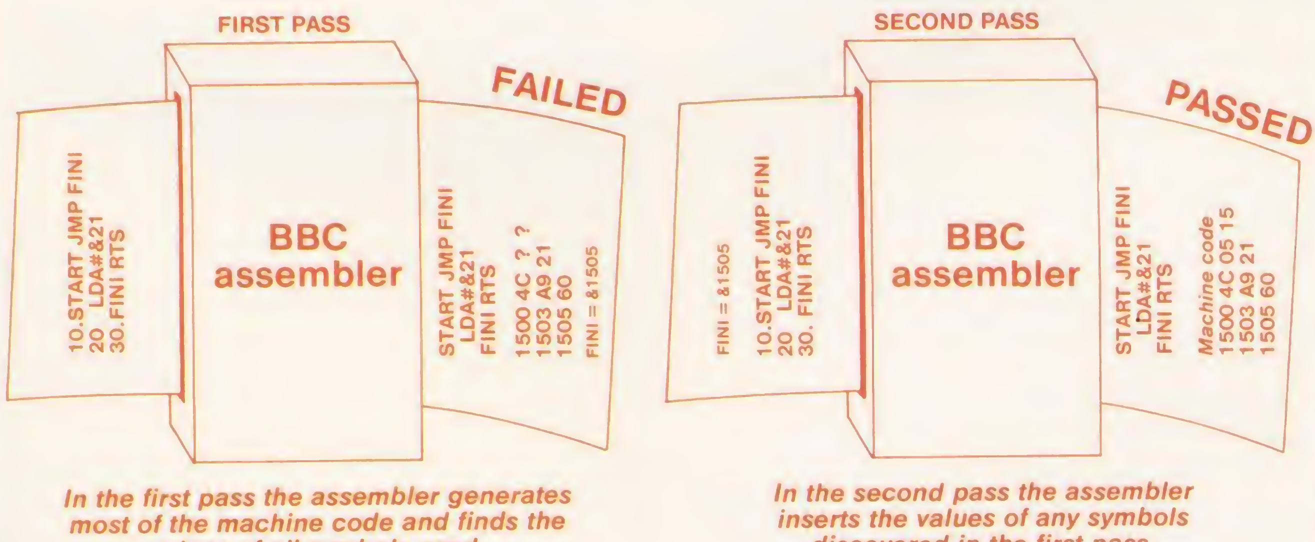


Figure 1. If a program contains symbols that appear before they are defined, a two pass assembly must be used.

However, careful examination of the machine code generated shows the value inserted for FINI is wrong.

The solution is to pass the program through the assembler a second time, figure 1. On the first pass forward references are 'noted' so that the assembler can insert the appropriate value for the label FINI in the second pass.

Initiating the two passes is a simple task. Since the assembler is part of Basic all that is required is to include the assembly language statements within a FOR ... NEXTloop. Also by allowing the counter variable 'PASS' to take the values 0 for the first pass and 3 for the second we ensure that error messages are suppressed only in the first pass of the assembler.

```
 5 FOR PASS = 0 TO 3 STEP 3
10 P% = &1500
20 [OPT PASS
30 .START            JMP FINI
40                   LDA#&21
50 .FINI             RTS
60 ]
70 NEXT PASS

listing
1500 OPT PASS
1500 4C 05 15        START JMP FINI
1503 A9 21           LDA#&21
1505 60              FINI RTS
```

Quite often in a Basic program a group of statements will be repeated at several places. To avoid this repetition these lines can be removed from the main program and placed in a separate subprogram called a subroutine. The GOSUB statement is then used in the main program to call the subroutine which in turn ends with a RETURN statement to return control at the completion of the subroutine to the main program.

Assembly language programs can also use a similar structure with a Jump to SubRoutine (JSR) instruction transferring control to the subroutine and a ReTurn from Subroutine (RTS) instruction returning control to the main program.

To help illustrate the use of these statements we will examine a short subroutine that creates a time delay. Many programming applications require the use of a time delay – eg to produce the flashing energy pills in *Snapper* or to govern the rate at which data is captured by the analogue to digital converter in a data logging application. Often these delays are formed using special purpose timer chips – eg 6522 VIA. However, it is possible to use a software loop that effectively wastes time by performing some useless counting operation. Figure 2 shows an example of this type of program. Its starting address has been given the name 'DELAY' and the routine ends with an RTS instruction.

Figure 3 shows an assembly language program that uses the delay subroutine to slow down the output of characters to the screen.

```
DELAY   LDX #255    /put 255 into X reg
LOOP    DEX         /decrement X reg
        BNE LOOP    /branch to loop if X reg not zero
        RTS         /return to BASIC
```

This routine provides a short delay by forcing the processor to continually go round a loop until such time as the X register contents are zero.

The X register is loaded with 255 (the largest number it will take) and the next instruction, DEX, reduces its contents by one. The following instruction, BNE LOOP, will force program control to execute the DEX instruction again and again until such time as the X register contents are zero. The precise effect of the BNE instruction is to examine the state of the zero flag of the status register and Branch to the specified address (LOOP) if the flag is not set at one. In the example this results in the continuous execution of the instruction sequence.

```
LOOP DEX
     BNE  LOOP
```

until zero is reached in the X register.

To lengthen the delay two loops might be used involving both the X and Y registers:

```
DELAY    LDY #255
LOOP 1   LDX #255
LOOP 2   DEX
         BNE LOOP2
         DEY
         BNE LOOP 1
         RTS
```

In this case the inner loop (LOOP2) is identical to the earlier program. By also using the Y register to count down and by reloading the X register value each time it reaches zero, much longer delays can be obtained.

Figure 2. A simple delay subroutine

Buried in the read only memory chip(s) that form the operating system controlling the Beeb are many useful machine code subroutines. Unlike some manufacturers, Acorn have released detailed information on these routines describing their function and how to use them (*User Guide* page 450).

Two routines that are simple to use and of particular interest are:

Name	Address	Function
OSDRCH	FFE0	Read character from keyboard into accumulator
OSASCI	FFE3	Write character from accumulator to screen

At first glance OSASCI appears to perform a similar task to our earlier

```
10 REM PROGRAM TO ILLUSTRATE ASSEMBLY
   LANGUAGE SUBROUTINES
20 SCREEN=&7C10 :REM FOR MODEL A USE &3C10
30 FOR PASS=1 TO 3 STEP 2
40 P%=&1500
50 [OPT PASS
60 .START LDA #65      \ASCII A
70        JSR DELAY    \TIME DELAY
80        STA SCREEN   \PLACE ON SCREEN
90        JSR DELAY    \WAIT
100       STA SCREEN+1 \SAME AGAIN TO NEXT
110       JSR DELAY    \THREE LOCATIONS
120       STA SCREEN+2
130       JSR DELAY
140       STA SCREEN+3
150       RTS          \RETURN TO BASIC
160 \TIME DELAY SUBROUTINE
170 .DELAY LDY #255  \DELAY SUBROUTINE
180 .LOOP1 LDX #255  \CONTROL WILL NOT
190 .LOOP2 DEX       \RETURN UNTIL ALL
200       BNE LOOP2 \LOOPS COMPLETED
210       DEY
220       BNE LOOP1
230       RTS          \RETURN TO CALLING PROGRAM
240 ]
250 NEXT PASS
```

a

```
>RUN
1500            OPT PASS
1500 A9 41      .START LDA #65      \ASCII A
1502 20 02 15   JSR DELAY    \TIME DELAY
1505 8D 10 7C   STA SCREEN   \PLACE ON SCREEN
1508 20 08 15   JSR DELAY    \WAIT
150B 8D 11 7C   STA SCREEN+1 \SAME AGAIN TO NEXT
150E 20 0E 15   JSR DELAY    \THREE LOCATIONS
1511 8D 12 7C   STA SCREEN+2
1514 20 14 15   JSR DELAY
1517 8D 13 7C   STA SCREEN+3
151A 60         RTS          \RETURN TO BASIC
151B            \TIME DELAY SUBROUTINE
151B A0 FF      .DELAY LDY #255  \DELAY SUBROUTINE
151D A2 FF      .LOOP1 LDX #255  \CONTROL WILL NOT
151F CA         .LOOP2 DEX       \RETURN UNTIL ALL
1520 D0 FD      BNE LOOP2 \LOOPS COMPLETED
1522 88         DEY
1523 D0 F8      BNE LOOP1
1525 60         RTS          \RETURN TO CALLING PROGRAM
```

b

```
1500            OPT PASS
1500 A9 41      .START LDA #65      \ASCII A
1502 20 1B 15   JSR DELAY    \TIME DELAY
1505 8D 10 7C   STA SCREEN   \PLACE ON SCREEN
1508 20 1B 15   JSR DELAY    \WAIT
150B 8D 11 7C   STA SCREEN+1 \SAME AGAIN TO NEXT
150E 20 1B 15   JSR DELAY    \THREE LOCATIONS
1511 8D 12 7C   STA SCREEN+2
1514 20 1B 15   JSR DELAY
1517 8D 13 7C   STA SCREEN+3
151A 60         RTS          \RETURN TO BASIC
151B            \TIME DELAY SUBROUTINE
151B A0 FF      .DELAY LDY #255  \DELAY SUBROUTINE
151D A2 FF      .LOOP1 LDX #255  \CONTROL WILL NOT
151F CA         .LOOP2 DEX       \RETURN UNTIL ALL
1520 D0 FD      BNE LOOP2 \LOOPS COMPLETED
1522 88         DEY
1523 D0 F8      BNE LOOP1
1525 60         RTS          \RETURN TO CALLING PROGRAM
```

Figure 3. Sample program illustrates use of a subroutine. (a) Gives code generated after the first pass, (b) gives final code after second pass. Note that the correct address for DELAY is obtained on second pass.

programs, transferring ASCII codes from the accumulator to the screen. However, its job is much more involved with it taking care of all the screen housekeeping duties, eg:

- placing a character in the next available screen location.
- taking a new line when a line is full of characters
- scrolling the screen when the screen is full.
- coping with the different screen memory maps in the different display modes.

To use the routine simply place the ASCII code for the chosen character in the accumulator and call OSASCI. For example:

```
10 REM PROGRAM PLACES "A"
   ON SCREEN
20 REM WORKS IN ANY MODE
30 OSASCI = &FFE3
40 P% = &1500
50 [
60    LDA #65    /ASCII for "A"
70    JSR OSASCI    /onto screen
80    RTS    /back to BASIC
90 ]
100 CALL &1500
110 END
```

Finally figure 4 gives a program that employs both these subroutines. OSDRCH is used to read a character from the keyboard into the accumulator. OSASCI then transfers the character to the screen. Effectively this turns the Beeb into a typewriter with each key depressed appearing on the screen.

To summarise there are many reasons why subroutines are useful. Just as with Basic the same subroutine can be called from different parts of the main program. Also, a subroutine developed for use in one program may later turn out to be useful within another quite separate program. In short, use of subroutines is likely to save the programmer time by eliminating the need to write from scratch, equivalent sections of program on more than one occasion. Furthermore, subroutines can be a considerable help when debugging a program especially if the programmer has ensured they are well documented and each performs a clearly identifiable task in relation to the problem.

Figure 4 on page 16 ▶

▶ from page 14

```
 10 REM PROGRAM TO TURN BEEB
 20 REM INTO TYPEWRITER
 30 REM
 40 REM FIRST DEFINE OPERATING
 50 REM SUBROUTINES
 60 OSASCI=&FFE3
 70 OSRDCH=&FFE0
 80 P%=&1500
 90[
100.START JSR OSRDCH \GET CHARACTER FROM KEYBOARD
110       JSR OSASCI \PLACE ON SCREEN
120       JMP START  \AND AGAIN
130]
140 END
>RUN
1500
1500 20 E0 FF .START JSR OSRDCH \GET CHARACTER FROM KEYBOARD
1503 20 E3 FF JSR OSASCI \PLACE ON SCREEN
1506 4C 00 15 JMP START  \AND AGAIN

    THIS IS A SAMPLE OF THE BBC TYPEWRITER PROGRAM.

    UPPER AND lower case POSE NO PROBLEMS.
```

Figure 4. Using operating system subroutines, with a sample of text using a typewriter program.

ALTERNATIVE DISCS

The lamentable lack of information on the disc system provided with upgraded machines can cause problems – especially as the interface uses up 2½k of the Beeb's memory. Joe Telford has some solutions, and a way round the shortage of BBC drives

During August I found that my two week holiday would be sufficient for my micro to have a disc interface fitted. I returned to switch on, and be greeted by a new opening prompt:

```
BBC Computer 32K
Acorn DFS
BASIC
>
```

At this point I was in the same position as anybody else with a new disc interface. My machine was returned without documentation of any changes, as this is supplied with the disc drive package. My first question was whether the additions would alter normal cassette handling. I fitted a cassette into the recorder, pressed rewind then stopped the recorder. I typed LOAD "" and started the recorder. Horror! The cassette LED refused to light and the micro just sat there. After a while it dawned on me that the machine must now power up in the disc filing system mode (DFS). Typing *TAPE12 brought the tape filing system into play. Now I could continue using my programs on tape. The rule became: Switch on... type *TAPE... then go. Pressing BREAK with the disc interface and DFS in place caused the micro to reset to the disc handling system, and had to be cured by including in my start-up sequence the direct command:

```
*KEY10 *TAPE12|M
```

This had the effect of setting up the tape filing system on BREAK.

Then, while running a rather large program, I was faced with the message Bad mode, where previously there had been no errors. I checked for free memory with the two direct commands:

```
DIM F%-1 : P.HIMEM-F%
```

and found only a few spare bytes when I expected to find over 2k. My first step was to see where the program actually began. Typing P.PAGE produced an answer of 6400. Prior to the upgrade, this had been 3584, which meant the upgrade had cost me 2816 bytes of RAM.

Thinking furiously, I realised the disc workspace from location 3584 (the old page) to 6400 (the new page) would only be used if I called DFS routines. The space was still available, provided I could convince my micro to use it. My fix was to load the program as usual at the new page location (6400), then relocate it to start at 3584. The necessary commands were allocated to function key 0 :

```
*KEY0 FOR J%=PAGE TO TOP:?(J%-
2816)=?J%:NEXT:PAGE=&E00|
MOLD| M*TAPE|M
```

The program is moved down in memory, then the routine sets PAGE to its OS 0.1 value, does an OLD to reset pointers to the program and finally sets up the cassette filing system. This also allows you to load from disc, and use F0 to move page and program down in memory together.

A program loaded and relocated will run as per normal, but any reference to the DFS will corrupt that part of a relocated program below location 6400.

Using this technique with discs fitted should still allow you to load and then relocate below the new PAGE, but once the program is loaded the user must switch to the tape filing system.

This expands the rules for using an upgraded system with cassettes to:

power on,
reconfigure the BREAK key,
configure function key 0,
LOAD program and relocate if necessary.

By this time I had realised that a number of alterations had been made inside my little micro.

First, the area marked for the disc interface had been fitted out with ICs, the most obvious being the disc controller – IC78, just above the keyboard connector on the main circuit board. Second, a small IC (IC27) near the power supply had a modification which connected one of its legs to a solder pad 4cms away. This sort of upgrade modification being enough to make the enthusiast think twice about doing it at home.

Next, the four EPROMs making up OS 0.1 had been replaced by two EPROMs marked V1.00 which were fitted into a carrier board, and finally a new chip marked DFS 0.97 was plugged into an empty ROM socket.

This was quite an exciting discovery, as it is rare to put a disc operating system (DOS) into ROM. The normal approach is to copy it from disc to RAM. What I needed to do on discovering the DFS chip was to find out what was in it. Experimenting a little, I typed *HELP which resulted in:

```
DFS 0.97
DFS
UTILS
OS 1.00
```

This confirmed my investigations under the cover. But what was UTILS? Typing *HELP UTILS

Figure 1. Comparison of signal names

Pin	Tandy signal name	BBC signal name
2	not connected	not connected
4	not connected	not connected
6	not connected	not connected
8	index pulse	index
10	drive select 0	drive select 0
12	drive select 1	drive select 1
14	drive select 2	not connected
16	motor on	load head
18	direction select	direction
20	step	seek step
22	write data	write data
24	write gate	write enable
26	track zero	track zero
28	write protect	write protect
30	read data	read data
32	drive select 3	side select
34	not connected	not connected

All odd numbered pins are tied to ground (both ends)

produced

```
DFS 0.97
BUILD <fsp>
DISC
DUMP <fsp>
LIST <fsp>
TYPE <fsp>
```

Encouraged by this I tried *HELP DFS which produced

```
DFS 0.97
ACCESS  <afsp> (L)
BACKUP  <src drv><dest drv>
COMPACT  (<drv>)
COPY  <src drv><dest drv><afsp>
DELETE  <fsp>
DESTROY  <afsp>
DIR  (<dir>)
DRIVE  (<drv>)
ENABLE
INFO  <afsp>
LIB  (<dir>)
RENAME  <old fsp><new fsp>
TITLE  <title>
WIPE  <afsp>
```

Most of these commands are easy to understand, provided the users has experience of discs. I was now faced with a system which had everything needed to operate disc drives, but no drives.

A friend was to hand with a pair of Tandy disc drives and a Tandy Interface handbook. I hurriedly compared signals from matching pins, on both BBC and Tandy disc drives. Figure 1 demonstrates the results. The connections were so close that I decided to attach the drives and experiment. The first step was to make up a suitable lead. Most drives seem to use a printed circuit board connector, and the Tandy was no exception. A suitable connector was found and a length of 34-way ribbon cable fastened into it. Figures 2 and 3 show the connector complete with cable. In my case there were two drive connectors as the drives were a twin pair.

The BBC end was more problematic. The connections in the *User Guide* on page 499 were more than a little incorrect when it came to the disc connector. I could only count 34 pins on my micro, rather than the 40 in the guide. In addition, the *User Guide* numbers the pins in a different order to my micro. Either I have a unique micro or . . .

Figures 4 and 5 show my speedbloc connections. Note that in figure 5 the ribbon cable is folded over on itself for reinforcement. The whole connector was placed in a vice and clamped. It is important to use a centrally polarized connector, which will fit the BBC's keyway, for the disc lead. Once set up, the lead was plugged

Figure 2. PCB connector at disc drive

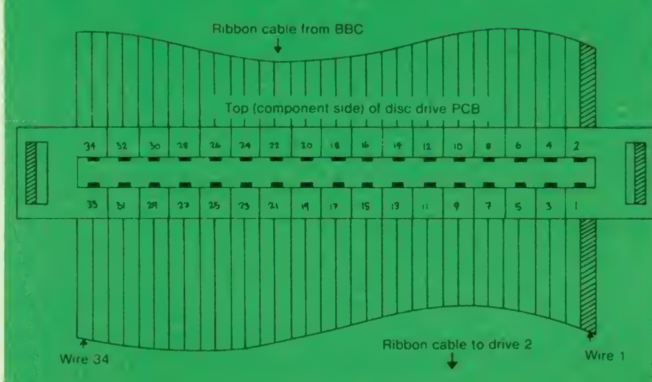


Figure 4. Format view of Speedbloc socket to connect to BBC disc drive port on micro

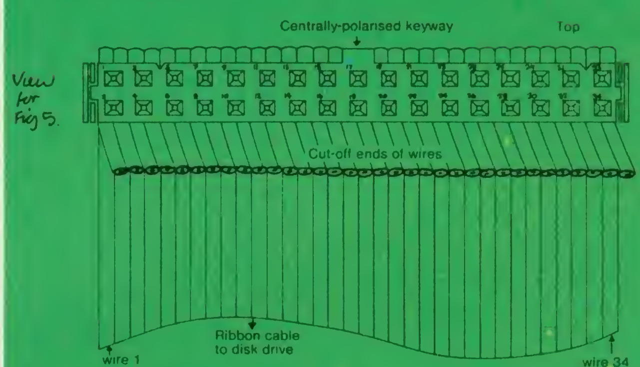


Figure 3. Side-view of drive connector

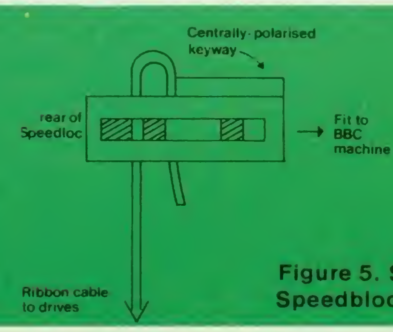


Figure 5. Side view of Speedbloc connections

in at both ends and the two devices switched on. Typing *CAT evoked a screenful of garbage (I had loaded a Tandy disc). Confident the combination would eventually work, I set about trying to initialise or format a disc. This important stage may be likened to the fitting of a margin and ruled lines to a blank sheet of paper. Unless a disc is formatted, the micro cannot use it effectively. What I needed was a program which would lay down 40 concentric tracks of 10 sectors each. The DFS chip did not contain such a program. This major hitch was explained away by Acorn who said their formatting program was included with the BBC disc drive package. Fortunately, help was at hand from the Tandy program *Superzap* which allows the user to format discs as he wishes.

I found out from Acorn that the BBC DFS expects to find its directory on the outmost track's first two sectors. *Superzap* puts the directory on Track 20. My next approach was to use *Superzap* to set every byte on the correct sectors to 0. Success. No garbage. Figure 6 shows the difference in layout between BBC formatted discs and Superzapped discs. *Superzap* really needs rewriting for the BBC machine.

Having proved I could indeed use an 'out of house' product on the BBC micro, I now needed to purchase same. Tandy discs were available ex-stock, but were more expensive than BBC drives. A local shop ordered a twin drive system from Cumana. The drives arrived marked 'For a TRS80', were plugged in and worked.

The disc drives have now run for three months. They are slightly larger, slightly cheaper, and slightly quieter than some BBC drives I have heard. They contain their own power supply, which I feel reduces the load on the BBC micro.

The result of this harassed week was that I could go back to considering the commands of the DFS chip. I am fortunate to have eventually been able to leaf through the BBC disc system manual which makes interesting reading (see commands on page 23).

Two wild characters exist on the DFS. The first is # which replaces any single character so that *INFO PROG# would produce information on files such as PROG1 PROG2 PROGB etc, provided they exist in the catalogue. The second is *. This replaces any group of characters, so that *INFO PRO* would produce information on files like PROG PROG1 PROG2 PROBLEM PROGRAM etc. A special combination *INFO*.* produces information of all files on a disc: ie *INFO (any directory).(any file).

In addition to the DFS commands above, LOAD SAVE *SAVE *LOAD *RUN *EXEC and *SPOOL operate as per normal on the disc system. *HELP is an operating system command. One useful facility sets *OPT commands for disc use. The commands available are:

*OPT 1,1 To enable the display of a files information as per *INFO whenever the file is accessed.

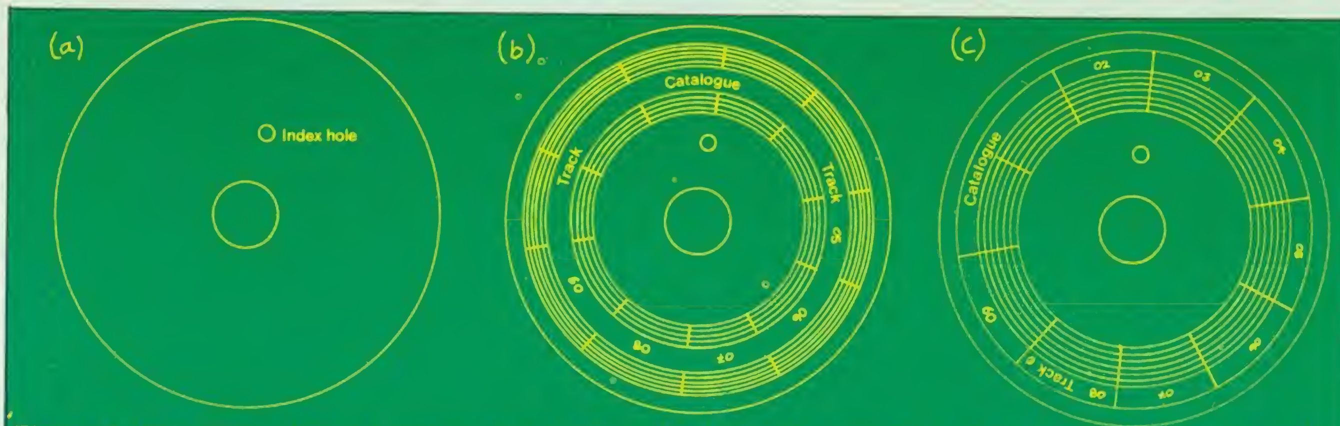


Figure 6. Disc formatting (a) Unformatted disc. Index hole is physical clue to start of sector 0 (b) Disc formatted by Superzap (c) BBC formatted disc. (Note: 40 tracks at 10 sectors per track numbered from sector 0 to 9, and track 0 to 39).

*OPT1,0 To disable the above.
*OPT4,1 To *LOAD a file called !BOOT (on BREAK or SHIFT–BREAK)
*OPT 4,2 To *RUN a file called !BOOT (on BREAK or SHIFT – BREAK)
*OPT 4,3 To *EXEC a file called !BOOT (on BREAK or SHIFT–BREAK)
*OPT 4,0 Cancels all other *OPT4s.

As can be seen, *OPT4,n is the start-up option for the disc system. Imagine that you as a programmer have to configure a system for a beginner to use. He simply wants to switch on and go, without learning any machine handling, such as LOAD. Let's call the program *Ledger*. The disc can be set up as follows:

Format and title a blank disc.
Set *OPT4,3
Save LEDGER on the disc.
*BUILD !BOOT
on receiving the line 1 prompt, type CHAIN "LEDGER" then press RETURN. Now press ESCAPE, to set !BOOT on the disc.

Whenever the user wishes to operate *Ledger*, all he needs to do is to switch on, place the disc in drive 0 and press SHIFT–BREAK. *Ledger* will start to run, having been called by line 1 of !BOOT.

One problem with the Acorn approach to its disc system is that each drive only accepts 31 files as the catalogue is on only two sectors in the disc. It is a problem mainly for users who write short programs, because the single-sided, single density drives used by most people at present, will only cope with about 10 by 10k programs per disc. However, having a 'catalogue full' message with half the space on the disc still available is annoying, to say the least.

It would be useful if we could simulate the auto-start facility of the Apple on the BBC micro. It would also be useful if a wider selection of drives could be made available for use on the Beeb. In fact both these features do exist, though they are not terribly well documented. There is a set of links on the front right hand side of the keyboard, which are normally unconnected

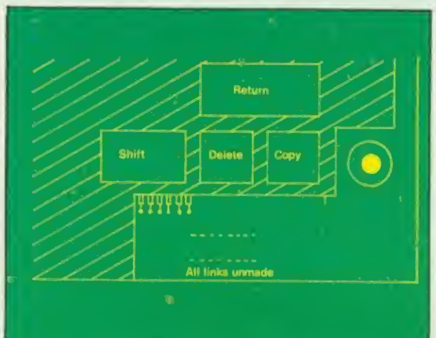


Figure 7. Keyboard links

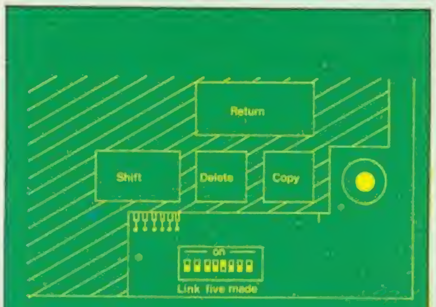


Figure 8. Keyboard links with DIL switch in place

Figure 9. Table of actions of keyboard links

Links:			Action
3	4		Disc drive selection
unmade	unmade		Olivetti: Tandy: Cumana
unmade	made		MPI drives.
made	unmade		Tandon: Shugart: 6 msec access
made	made		Tandon 4 msec access time
5			Auto boot selection
unmade			by pressing SHIFT-BREAK
made			by pressing BREAK
6	7	8	Screen mode on BREAK
unmade	unmade	unmade	mode 7
unmade	unmade	made	mode 6
unmade	made	unmade	mode 5
unmade	made	made	mode 4
made	unmade	unmade	mode 3
made	unmade	made	mode 2
made	made	unmade	mode 1
made	made	made	mode 0

(figure 7). Making soldered links here will eventually ruin the printed circuit tracks if they are continually soldered and unsoldered. Figure 8 shows an eight pin DIL (dual-in-line) switch costing £1.50 which I connected to ease the problem of making and breaking the links. There are four types of effect demonstrated by making these links (setting the switches to ON).

- Switches 1&2 are not yet assigned any function.
- Switches 3&4 select the type of Disc drive in use.
- Switch 5 selects the form of Automatic startup (Auto Boot)
- Switches 6,7&8 select the screen mode on power up or BREAK.

Figure 9 gives more detailed information on these links.

That's it for January. Next month back to software . . . dynamic procedures – naughty (say Acorn) but nice.

DFS and UTILS commands (alphabetically)

*ACCESS <afsp> (L)
Locks or unlocks a file eg,
*ACCESS SPIRAL L locks the file 'SPIRAL'
*ACCESS SPIRAL unlocks it.
A locked file cannot be saved to, deleted, wiped, renamed, or destroyed. It can of course be formatted.

*BACKUP (src drv) (dest drv)
Allows a disc to be copied in total. It needs to be enabled eg,
*ENABLE
*BACKUP 1 0
will copy all information from drive 1 onto drive 0.

*BUILD <fsp>
Creates an ASCII (text) file directly from the keyboard.
*BUILD MESSAGE
will display line numbers to let the user enter data in a primitive text editor. Once the ESCAPE key is pressed the file MESSAGE is placed on disc where it may be accessed via *EXEC *LIST or *TYPE.

*CAT <drv> gives a catalogue of a disc on the screen.
*CAT 1 gives a catalogue of drive 1
*CAT or
*. catalogues the current drive (set by *DRIVE).

*COMPACT <drv>
tidies the disc in the named drive so that all the space is moved to the end. This can help cure DISC FULL messages eg,
*COMPACT 1 will tidy the disc in drive 1.
The command has a habit of using RAM in the BBC micro – overwriting any current program.

*COPY <src drv><dest drv> <afsp>
allows a file to be copied from one drive to another or from one drive to itself.
*COPY 0 1 WHIZZ copies WHIZZ from a source disc on drive 0 to a destination disc on drive 1. Again, user RAM may be overwritten

*DELETE <fsp>
To directly remove an unlocked file from the catalogue of a disc. A deleted file cannot be revived.
*DELETE BOMBER will remove Bomber from current disc in use.

*DESTROY <afsp>
removes all unlocked files specified. Needs *ENABLE eg,
*ENABLE
*DESTROY *.*
will remove all the files from the disc, after asking permission first. Again, destroyed files cannot be revived.

*DIR (<dir>)
Changes the current directory name to a single character. Set on power up to
*DIR C
This sets the current directory to 'C'. Any saved file will be prefixed with C.

*DRIVE <drv>
selects the current drive in use eg,
*DRIVE 1 sets the current drive to 1 so any command not specifying a drive will log onto drive 1. Such commands may be
*CAT, LOAD, SAVE.

*DUMP <fsp>
produces a listing in hex of the file on the screen eg,
*DUMP SPIRAL will produce a hex listing of SPIRAL. (Not necessarily machine code.) The use of CTRL–N will select paging mode, which will greatly help readability.
*ENABLE
allows the irreversible commands *BACKUP and *DESTROY to be used. It is a safety lock.

*INFO <afsp>
displays the following information on the files specified: directory, filename, access, load, address, execution address, length in bytes, start sector eg,
*INFO *.* displays the above information on all files on the current disc.

*LIB (:<drv>.)<dir>
sets the library to the specified drive and directory eg,
*LIB :1.C sets the library to drive 1, directory C.
The library facility allows extensions to the * commands. A user could place his own machine code routines on disc and call them by name. Acorn use three such programs *FORM40 *FORM80 and *VERIFY on their master disc, included with the BBC drive.

*LIST <fsp>
displays an ASCII (text) file created by *BUILD, complete with line numbers eg,
*LIST MESSAGE will produce a line numbered listing of MESSAGE. LISTing program files in this way is not suggested, as unforseen errors occur.

*RENAME <old fsp><new fsp>
changes the name of a file eg,
*RENAME WHIZZ BUZZ changes WHIZZ to BUZZ on the current disc.

*TITLE <disc name>
retitles the disc with the chosen name – up to 12 characters long eg,
*TITLE "MASTER DISC" will display the words MASTER DISC on future *CATs of the disc.

*TYPE <fsp>
displays a textfile on the screen without linenumbers eg,
*TYPE MESSAGE will display the contents of MESSAGE as a text file. The use of CTRL–N is again recommended.

*WIPE <afsp>
removes specified files from catalogue, asking permission over each file. Unlocked files only can be removed. WIPEd files cannot be revived eg,
*WIPE *.C* wipes all files beginning with C.

Key to DFS command parameters

fsp	file specification
afsp	alternative file specification
src	source
drv	drive
dest	destination
dir	directory letter

File specifications

A complete file specification consists of
: drv . dir . filename
eg *EXEC :1.C.SPACE
will exec the file SPACE on directory C of drive 1
and *EXEC SPACE
will exec the file SPACE on the currently selected directory of the currently selected disc. Note that drive and directory can be selected with the appropriate commands.
Commands LOAD and SAVE must have the file specification enclosed in quotes eg,
LOAD":1.G.DT1" or LOAD"DT1".

POWER over your BBC micro is what Ian Birnbaum gives you in this regular series. By answering questions and developing readers' ideas, he will increase your control and understanding of the techniques of programming. At the same time he can reveal the quirks of the Beeb so you can put them to good use.

Readers who send in ideas will be rewarded with:

£5 for each tip printed,
£10 for good hints,
£20 for exceptional ideas.

So, whether you have a problem or a tip, send it to: *BBC Forum, Acorn User, 53 Bedford Square, London WC1B 3DZ.*

He cannot answer every letter personally, but will return contributions if a stamped addressed envelope is provided.

WRITING A PROGRAM TO ERASE PART OF ITSELF

One handy feature of BBC Basic is the ability to mix assembly language and Basic freely. In particular this allows the use of machine code routines in those parts of a program where speed is important.

On the standard (non-disc) machine, room is reserved from &D00 to $DFF to put your machine code routines. Since this space is not part of the Basic memory, which runs from &E00 to HIMEM, you can use machine code routines without reducing space available for Basic.

However, the advantage is lost if you have to include the assembly language in your program. Assembly language occupies five times as much space as machine code, and usually more. One possibility is to dispense with assembly language altogether, and instead save the machine code with the Basic. *SAVE "Program" 0D00 ???? (where ???? is TOP in hex) will do this.

This solution is not perfect, however. The user needs to *LOAD the program; and, what is worse, needs to remember to type OLD (or END or LIST, or press ESCAPE) before running it. There is a better solution: put the assembly language at the end and, after it has assembled into &D00 onwards, erase it! This is feasible because once the machine code translating has been done, the assembly language has no further function. But how do we erase part of the program from inside itself?

Suppose the assembly program will go from line 2000 onwards. The first line, 10 say, should read:

```
10 GOSUB 2000: CLEAR
```

At line 2000 write:

```
2000 GOSUB 2010: HIGH = TOP–XXXX: ?&12 = HIGH MOD256: ?&13 = HIGH DIV256: HIGH?–1 = &FF: HIGH?–2 = &0D: HIGH?–3 = &F8: HIGH?–4 = 5: RETURN
```

where 2010 onwards contains your assembly language which will assemble into &D00 onwards (or some other protected location). The four Xs will contain a specific number; which one, will be explained shortly.

When you run your program, lines 2000 onwards will be erased after the machine code is assembled, and line 2000 will now read:

```
2000 RETURN.
```

Thus, whenever the program is subsequently run, things will operate normally, but the program will occupy less space. (Of course, any attempt to save the program once it is run will lose the vital assembly language.)

Let's look at how this works, now: we will then be in a position to

VALUABLE VARIABLES

In November's issue I gave a method of programming the BREAK key so that the values of variables are not lost. The same idea can be used for editing. It is often desirable, when debugging, to be able to change lines where bugs occur whilst retaining the value of variables set in the program. Then you can go to the relevant section and continue debugging with the old values intact. Let's see how to accomplish this.

The first thing to do is to set line 0 to read

```
0 LOMEM=TOP+1000
```

(assumes line 0 is otherwise unused in your program).

This should save enough space above the program for any changes you make or new lines you add. Then program key 0 and key 1 as in listing 1.

When a bug occurs, or you wish to make a change in your program, press key 0. This will save the variable table plus the address of the first free dynamic variable location in &D00 to &D73. Make all your changes and delete line 0. Then press key 1, and the information saved in &D00 to &D73 will be put back into the correct place. Now re-enter your program with the relevant GOTO (but do not use RUN, of course, since all variables will be lost).

Listing 1.

```
*KEY0 !&70=I%:!&D00=!2:FORI%=0TO115:I%?
&D02=I%?&482:N.:I%=!&70|M
>*
>
>*KEY1 !&70=I%:?2=?&D00:?3=?&D01:FORI%=
0TO115:I%?&482=I%?&D02:N.:I%=!&70|M
```

PROGRAM PROTECTION

Here is an easy way to protect a program against copying. Consider this simple program in listing 2. Run it and note the value of TOP.

Now type in listing 3 (the underlined section is the computer's response) and save the program on tape.

The program can now be LOADed in the normal way (but cannot be CHAINed). When loaded, the message 'Bad Program' will appear. The program will not LIST and cannot be SAVEd (it can, of course, be *SAVEd). However, the program will RUN in the normal way, and what is more, TOP will retain its correct value. Hence to reclaim the program type ?(TOP-1)=255 and then LIST.

If you use END instead of STOP at line 70, when the program ends the message 'Bad program' will appear, and TOP will no longer have its correct value. This is not a disadvantage as long as you remember the correct value of TOP. Indeed, it can be an advantage if you wish to make it more difficult to list and copy the program. To make it more difficult still, vary the value of ?(TOP+1).

Listing 2.

```
10@%=1
20A$="OK":A=1:B=2:GOTO50
30PRINTA$;B
40@%=2570:GOTO70
50PRINTA$;A
60GOTO30
70PRINT~TOP:STOP
```

Listing 3. Underlined part is computer's response

```
>?(TOP-1)=0:?(TOP+1)=2
>P.~TOP+2
      E5B
>*SAVE "TEST" 0E00 0E5B
```

fill in the four Xs in line 2000 for any particular application. The value in HIGH needs to be the value TOP would take if 2000 consisted solely of RETURN and if there were no further lines after 2000. Locations &12 and &13 contain the value of TOP, and hence the low and high bytes of HIGH are put in here. Finally, the end of program marker is put where it should be if 2000 is the last line – one below HIGH – and line 2000 is configured to contain RETURN only (the token for RETURN is &F8).

When return is made from 2000 to line 10, a CLEAR is executed. This resets the first free dynamic variable location above Basic to be equal to TOP: but, of course, TOP is now HIGH. Hence all lines after 2000 will be overwritten by variables and line 2000 itself reads RETURN. This is exactly what we require.

The only problem remains how to fix the value of the four XXXXs. This should be done only when lines 2000 onwards have been finalised and completely debugged: the rest of the program could subsequently be changed, however.

Find the value of TOP (call it T%) when the entire program including

Bits of the program can be lopped off if not required

the assembly language, are in memory. Next save all the lines from 2000 onwards (the assembly language plus modifications) on tape. To do this find the hex for 2000 by using P.~2000. Then write:

I%=TOP: REPEAT: I%=I%-1: UNTIL (?I%=&uv AND I%?1=&wx): P.~I%-1 (where uvwx is hex for 2000).

Now write *SAVE "TEMP" abcd efgh (where abcd is I%-1 in hex and efgh is TOP in hex).

Then delete everything after line 2000, and replace line 2000 by 2000 RETURN. Find the value of TOP again. Then, XXXX = T%-TOP. If this is less than 1000, some of the remaining Xs will have to be replaced by spaces. You can now restore your program by executing *LOAD "TEMP". Insert the correct values for XXXX and resave the entire program.

This method of self-erasure has other applications: DATA statements could be erased after they have been read; procedures and functions could be erased if they are to be used in an early part of the program only; and, quite generally, bits of the program can be lopped off if no longer required in the present run. Of course these applications, unlike the assembly language one, require that the program is only going to be RUN once.

COMPUTERS – NOT JUST ANOTHER FAD

£9 million of taxpayer's money is being pumped into primary schools to provide computers – and yet many teachers have failed to master a video machine. This series sets out to destroy the mystique of computing. In 12 monthly instalments it will provide practical advice to teachers, and show parents what can be done in the classroom.

In July last year, the Government announced a £9m scheme to get a microcomputer into every primary school by the end of 1984. This took effect from October 1 and the Department of Industry is providing funds to allow each school to purchase an approved computer at half price.

Teacher reaction to this objective is mixed. There are enthusiasts who can't wait to get a computer and begin programming, but most teachers are more apprehensive. There is a widespread fear that this is just another fad; that in a couple of years micros will be relegated to the store cupboard as expensive toys along with other equipment such as the overhead projector or Synchrofax machine.

To allay these fears and allow micros to fulfil their undoubted potential, teachers must become more aware of the contribution which the micro can make to education. This article is the first of a series designed to promote such awareness. It will present a general framework which will be explained in depth by future articles. Contributions will come from 'experts' in educational computing and classroom teachers, all of whom are convinced of the value of at least one of the many applications of micros.

INTRODUCING AN EXCITING TOOL – THE COMPUTER

Heather Govier, a microelectronics advisor for the London Borough of Croydon sets the framework for this major new series

The idea behind this article is to present a view of the micro as an exciting tool, and the primary school as a favourable environment for its optimal use. Future articles will underline this view by exploring a wide range of applications in greater depth. There will be a concerted attempt throughout to demystify computers and explain some of the most pervasive jargon.

Articles on the use of the micro for mathematics and language development will be included. The first of these will examine the way micros can help the busy primary school teacher to achieve the six teaching styles advocated in paragraph 243 of the Cockcroft report. The information technology revolution will be featured in an article which considers what children should know about computing while another will discuss the philosophy behind teaching programming and the various approaches. Data handling and word processing will be explored more fully, as will the use of simulations. One article will consider the machine itself, with an evaluation of some alternatives to the standard keyboard input and screen output. The use of the micro in school administration will be covered, and in the next article a practising teacher will pass on his experience of organising the classroom and school timetable to make maximum use of the new technology.

The role of the computer in helping to introduce science and technology into the primary school will complete a series which will serve to heighten teacher awareness of the micro's potential in the primary school.

BOOSTING THE CURRICULUM

If the micro is to avoid becoming an expensive novelty, its role in the curriculum must be carefully established. The curriculum must be central, not the micro. Teachers should be asking: 'What can a computer do fo my school, my pupils, my teaching?'

This approach starts from the existing curriculum, seeks out its weak points and asks whether micros could help strengthen the curriculum. When viewed in this way the micro has three roles.

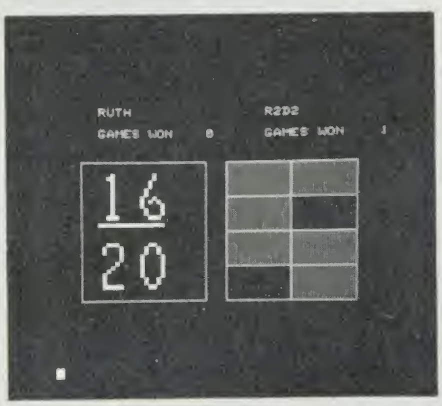
First, it can provide a novel approach to enliven the traditional curriculum, making teaching and learning more fun and more stimulating. Second, it can allow teachers and pupils to do things which would otherwise have been difficult or impossible. Third, it can extend the curriculum into the little-explored fields of problem-solving and logical reasoning.

There can be no doubt that most children enjoy working with computers. Fears that this fascination results from novelty have not been realised in schools where micros have been in use for some years. The micro can provide a stimulus for the poorly motivated pupil, making much of the traditional curriculum more enjoyable.

Programs which allow pupils to practice arithmetical skills or test their spelling abound. These vary in quality, but there can be no doubt that some are valuable educational tools for certain pupils. Imparting the basic skills to slow learners can be an uphill struggle. Much repetition is necessary which can be tedious for the pupil and frustrating for the teacher. Micros can make this learning much more pleasurable. Pupils who are slow to record answers can solve 30 sums at the keyboard in the time they would normally take to do three with pencil and paper. When calculations are incorporated into a game, essential practice becomes fun. Tasks can be completed at the pupil's own pace with no danger that the computer will become irritable or impatient.

The better programs can do much more than this. One example is a game which allows the pupil to write and solve subtraction sums. In addition, there are various strategies incorporated in the game which have to be worked out. Thus a simple practice exercise can be built into something which requires quite a lot of mathematical thinking.

The computer presents opportunities for pupils and teachers to take some aspects of the curriculum a little further. One such



Ruth takes on a computer christened 'R2D2' in a maths program

application is as a word processor. Pupils may type in a piece of writing, poetry or prose which can then be taken to the teacher for correction. Changes in spelling, punctuation or grammar can be made, and the opportunity taken to discuss and enrich vocabulary.

The pupils can make these changes at the keyboard without having to rewrite the passage.

The micro and the printer are able to free pupils from the onerous task of rewriting, allowing them to concentrate on the more important creative aspects. The final piece of work can be stored on cassette or disc for future use, for example in a school magazine. Using the word processing capabilities of the micro pupils could easily produce such a publication without secretarial support.

The micro allows pupils access to extensive information files such as census data. The computer is able to manipulate data in various ways, providing answers to such questions as: Where were the birthplaces of the people living in Victoria Street in 1880 who were over 50 years of age?

Simulations can also be used, allowing pupils to conduct experiments or explorations which would be impossible in reality. A typical example is a program which allows pupils to make an archeological study of an ancient Egyptian site. Pupils plan their expedition and choose a site for their dig. The computer supplies information about what they have found or failed to find, keeps a record of their financial position and presents problems which have to be faced and overcome. Data in the program is based on fact giving pupils a taste of what real exploration is like.

Word processing and information retrieval highlight the need for computers in primary schools to have access to disc storage rather than the cassette systems provided by the DoI Scheme. To handle large quantities of text or data also assumes that a printer is available. Otherwise valuable time is spent copying from the screen.

Use of word processors, databases and simulations still don't take the computer outside the existing curriculum. These applications take the tedium out of preparing text or sorting through information, allowing more refined text to be produced or more complex data analysed.

While the traditional primary school curriculum keeps pupils busy and indeed may teach them a

great deal, it rarely requires them to think deeply. Many pupils resent being made to reason and question, preferring the soft option of routine activity. With the micro the little explored fields of problem solving and logical reasoning can be emphasised.

One type of program which can foster this activity is the archeological simulation described. Pupils are invited to make conjectures about what would happen if ... and to test these conjectures through the program.

Another use is in mathematical problem solving. An example is a program which allows pupils to investigate the mathematics of the Spirograph. This toy consists of a number of cog wheels and rings which can be used to draw a variety of flower-like designs. There are interesting mathematical relationships between the number of teeth on the wheels and the number of petals drawn on the flowers. Using the computer in conjunction with the Spirograph tool pupils are able to set up mathematical hypotheses about these relationships, test them and accept, reject or refine them as appropriate.

Perhaps the most exciting use of micros in primary classrooms is the use of the turtle graphics instructions of the Logo language. These introduce pupils to some of the logical problem solving skills involved in computer programming. Instructions control the movements of a 'pen' around the screen and are used to draw pictures and patterns ranging from simple squares to complex spirals. Good versions of Logo encourage the habit of dividing problems into solvable sections, and the instant response of the turtle graphics means that lapses in reasoning become immediately apparent.

Use of Logo introduces something quite new into the primary school curriculum and its value is as yet unassessed. However, proponents of Logo claim that it could revolutionise the educational process and contribute substantially to the thinking of pupils.

The computer then can be used to enliven, enrich or extend the traditional primary school curriculum.

The first application may be a necessary first step for most teachers if they are to become familiar with the new technology. In the security of traditional curriculum areas they will be able to meet micros more easily. However, if the micro is to have a continuing role in education teachers must think beyond drill and practice to more creative uses. For this to happen, extensive and carefully planned in-service training is needed to help teachers become familiar with the machinery and programs available, while widening their horizons to appreciate the possibilities.

Initiative for such training could come from Government. Central funds have been made available to support schools in the purchase of hardware. If this hardware is to be used effectively it must be backed by appropriate training, provided in the educational budget.

GET THE BEST FF

- **FIND out what support is available from your LEA.**
- **TRY to find a place on a training course *before* the computer arrives.**
- **JOIN any local groups for teachers who use the same machine or share your educational philosophy.**
- **DON'T worry if you are no good at maths or can't program in Basic.**
- **CONTACT your local MEP regional information centre to find out where good programs can be seen. Don't buy programs unseen.**
- **JOIN MAPE, the organisation devoted to computers in primary schools.**

FROM YOUR MICRO

- **DEMAND high quality workmanship from suppliers, and don't make children struggle with faulty equipment.**
- **FIND somewhere safe to store your computer.**
- **KEEP equipment out of sight at night, and mark everything.**
- **DON'T take your computer home without checking the LEA rules. Contact Muse, Freepost, Bromsgrove, Worcs, for a good insurance scheme**
- **ORDER blank cassette tapes and buy programs before the computer arrives.**

Paul McGee
Inspector for Maths and Computing Croydon

THE LINK TO MATHS MAY BE BROKEN IN PRIMARY SCHOOLS

The primary school environment is a perfect setting for computer activity, in many ways better than the secondary area where educational computing began. Arguably, micros should have been introduced first in primary schools. The organisation of many primary schools means that subject boundaries are less distinct as one teacher covers most subjects. The curriculum is largely child-centred rather than subject-centred because the primary school is free of pressure from examination syllabuses or employer's demands. These factors enable the teacher to employ a variety of teaching styles and methods of classroom organisation, which provide major opportunities for the intelligent use of computers.

Because they are not subject specialists, primary teachers are better placed to perceive the potential of computers for applications across the curriculum. Had micros been introduced into the primary sector first it might have been possible to avoid the artificial association of computers and mathematics which developed because computers were first used in mathematics departments. Consequently, the mathematics staff were the first to become involved in writing programs for schools and gain the necessary experience and interest to proceed. A circular link has been set up which has proved very hard to break.

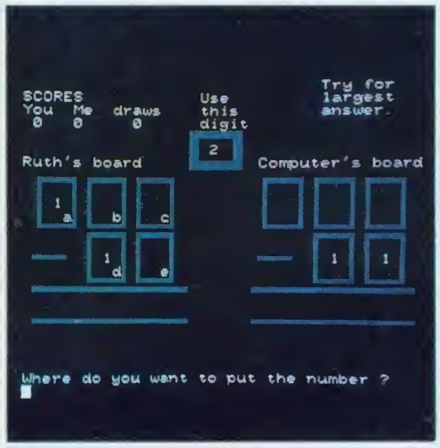
From this association has developed the myth that computers are mainly for boys. The long standing and well documented rejection of mathematical and scientific subjects by girls is wrongly transferred to computer work. This masculine image of the computer extends also to teachers. Despite the fact that most primary teachers are women, the majority of primary teachers attending courses on micros are men.

Such stereotyping might have been reduced had micros been introduced with more emphasis on their potential for language development and humanities. Later articles will show that this emphasis has been generated naturally in many primary schools. The current Government initiative and funds could present opportunities for breaking out of these restrictive cycles if appropriate training support were given.

In the primary sector there is no need to set aside a distinct slot for computer studies. The computer can infiltrate the whole curriculum with time and space found for some of the more innovative and exciting applications. Emphasis in primary schools must be on teaching and learning *with*, and not *about*, the computer. Technical details are best left to courses at secondary level.

It is far better that primary schools should be using micros as tools to enhance a topic-based approach to education. Junior pupils, however, should also know about some of the ways in which computers are used in the outside world. Word processing and simulations are all common uses made of computers in society. Pupils should have the opportunity to use micros themselves for all these and to appreciate the substantial differences.

There is still in the minds of many teachers and parents a 1984 image of serried ranks of pupils each sitting at a computer, working in total isolation. This must be dispelled. Computers need not be isolators, in fact a micro can be a valuable focus for group interaction. Again teachers in primary schools are well placed to optimise the potential of the micro because they are accustomed to using a variety



The enormous range of software can help children in all subjects

of teaching styles. No longer is all teaching chalk and talk, with the teacher addressing the whole class. This style is only one of three commonly-used approaches: class, group and individualised teaching.

Micros can be used with all three styles. At class level, a computer attached to a large television screen can be used for demonstrations. One example is a program which draws a graph of changes in the water level in a bath as taps are turned on or off, the plug put in or pulled out, or a man gets in or out. An understanding of a graph as a picture which tells a story can be gained by very young pupils in a way not possible without the moving images displayed by the computer.

At group level there are a number of programs designed to stimulate discussions and interaction. The *Animal* program asks the pupils to think of an animal and uses a branching series of questions to try to guess what they thought of. Questions such as 'Is it a mammal?' can generate purposeful discussion and may call for the use of reference books. Ultimately the program either guesses the animal correctly or asks for a question to distinguish the animal from the incorrect guess, stimulating pupils to think of a question to differentiate between a giraffe and an elephant, for example.

An important feature of this and many other programs is the ability to grow to accommodate the pupil's increased understanding and confidence. This can be uncomfortable for teachers who wish to treat pupils as receivers of accepted wisdom, but exciting for those who wish to exploit pupil's powers of creativity and logical thought. After using *Animal* one child claimed that it had taught her that computers didn't know everything – sometimes you can know more than they do!

Finally, there are a multitude of programs which test skills or

'No longer is all teaching talk and chalk'

present opportunities for problem solving by individuals. A typical example is *Jugs* which sets up problems for pupils to solve. The user is given two ungraduated jugs of given capacity on the screen and asked to measure out a quantity of water by pouring from one jug to another. For example, the jugs may hold five litres and seven litres and the pupil be challenged to measure accurately four litres. The problem can be solved by filling the seven-litre jug and pouring the contents into the five-litre jug until it is full. This leaves two litres in the larger jug. If the contents of the smaller jug are disposed of, the two litres can be poured into it, leaving a capacity of three litres. The seven litre jug is then filled and three litres poured into the five litre jug leaving four litres in the large jug.

The program allows pupils to manipulate the jugs, exploring possibilities freely and seeing the result of such explorations pictorially. Experience suggests that in the early stages of work with the program there is much random pouring and a fairly intuitive approach to the solution. However pupils' thinking can be channelled by a good teacher enabling the development of sound problem solving techniques. Such an activity needs quiet thought and concentration. It is ideal stimulation for a bright pupil and clearly allows an individual approach.

Many individual programs lack the range of responses needed to effectively teach difficult concepts but their motivational value can be high and they can free teachers from supervising pupils. Many good programs prevent pupils from practising errors, a fault in much text book work, and record the errors made for the teacher to examine.

Using computers is the same as using any other teaching aid or method. A good teacher will find the appropriate use for the pupil or group of pupils and never be bound by thinking a program can only be used in one particular way. Similarly, other teachers will use inappropriate programs, or use programs inefficiently. The same criticisms can be made of text books, work on the blackboard or talking by teachers or pupils.

SPECIAL READER OFFER

BBC MICRO INSTANT MACHINE CODE!

The Ruston BASIC Compiler on cassette, complete with detailed documentation in accompanying book offered to readers of *Acorn User* at a special discount price of:

£29.95 ~~£34.95~~

INSTANT MACHINE CODE FROM A GOOD SUBSET OF BBC BASIC?
Yes, it is true—type the BASIC program into your Model B BBC Micro, trigger the compiler, and your program is converted almost instantaneously into superfast machine code.

How fast does your BBC Micro run this program?

```
10 TIME=0
20 for T%=1 to 60000
30 NEXT T%
40 PRINT TIME
50 END
```

Unless you're already the proud owner of a copy of the *Ruston BASIC Compiler*, your BBC Micro will take 13.92 seconds to execute the program. With the *Ruston BASIC Compiler*, execution time is reduced to an insignificant 0.63 seconds (that's 4.5% of the original time).

Your own programs can show a similarly dramatic speed improvement by applying the Compiler to them. The only constraints are that some features of BBC BASIC are not supported (see below). The Compiler operates by examining the program to be processed (it can be on cassette/disc or in memory) and working out the fastest possible combination of machine code instructions necessary to achieve the same effect. This machine code can then be run in the same way as the original BASIC program.

The emphasis throughout the Compiler has been placed on superfast execution—unlike many compilers available for other computers. The end result is ideal for many applications: games, real-time monitoring and control, utilities and simple (but fast) arithmetic.

In addition, many useful features are added to the computer by the Compiler, enhancing the BBC Micro's position as the most versatile and powerful small computer on the market. These extensions take the form of a special 'sprite' graphics mode, a command to wait for vertical sync (vital for animation) and several others. 'Sprites' are objects which can be moved at will on the screen. You can define whether one will pass in front of or behind any others already on the screen. Moreover, as a sprite moves across the screen, the background is automatically restored. All this is done in a fraction of the time it would take in BASIC.

The *Ruston BASIC Compiler* is issued with a comprehensive manual. The Compiler is marketed by Interface who are confident that no other programm for the BBC Micro is as well documented.

Compiler keywords: ADVAL, AND, CALL, CLG, CLS, COLOUR, DRAW, ELSE, END, EOR, FOR, GCOL, GET, GOSUB, GOTO, IF, INKEY, INPUT, LET, MODE, MOVE, NEXT, OFF, PLOT, POINT, PRINT, REM, REPEAT, RETURN, RND, SOUND, TAB, THEN, TIME, TO, UNTIL and VDU, plus extensions! 26 integer variables are implemented (range 0–65535). The *Ruston BASIC Compiler* has been tested on all current versions of the operating system, and is proven with the BBC Disc System.

This valuable product is written by BBC Micro expert **Jeremy Ruston**—author of *The BBC Micro Revealed* (see below) and *The Book of Listings* (with Tim Hartnell).

ORDER FORM to Special Reader Offer, Acorn User, 53 Bedford Square, London WC1B 3DZ.
Please send me:

INSTANT BBC MACHINE CODE............................ copies at £29.95

Let Your BBC Micro teach you to program................ copies at £6.45

The BBC Micro Revealed............................ copies at £7.95

Total ________

I enclose my cheque for £............................ (payable to Interface)
or
Please debit my Access/Barclaycard/Visa/American Express/Diners Club

Account No. ☐☐☐☐☐☐☐☐☐☐☐☐☐

Signature............................ Date............................

Name............................

Address............................

....

Also available at list price are:

Let your BBC Micro teach you to program, Tim Hartnell, 194pp, £6.45. This book, by bestselling author Tim Hartnell, is the ideal companion for you if this is your first computer. It takes you step by step, through programming in BBC BASIC, with a number of worthwhile programs.

The BBC Micro Revealed, Jeremy Ruston, 144pp, £7.95. When you have mastered the contents of the BBC Micro Manual this book will enable you to continue your explorations of the computer's functions and capabilities. If you are serious about developing your programming skills this book is an indispensable aid " . . . destined to become the bible of the BBC user . . ." *(Personal Computing Today).*

COMMODORE HOOK-UP

The interface to Pet printers doesn't match the Beeb's, but Tim Edwards can overcome that problem

If you have a printer with a serial interface or a Centronics parallel interface, linking it up to a BBC micro is straightforward. However, a number of users have access to a Pet printer which uses an interface kown as the IEEE488 bus. Such users will be disappointed to hear that the BBC micro does not have an IEEE488 bus facility. Notwithstanding, this article shows how you can actually run a Pet printer from a BBC micro.

The solution consists mainly of a machine code routine to handle the IEEE488 protocol and replace the printer driver routines within the BBC machine's operating system. Some thought must also be given to the physical interconnections as it is necessary to use both the normal printer port and the user port of the 6522 versatile interface adapter (VIA).

The IEEE488 general purpose interface bus is a complex and versatile system by which a number of different devices can be connected together on a set of parallel cables, so that each can 'talk' to other devices on the bus. One of the devices is defined as the controller and decides who should talk to whom and when. At any time, it can hand over control to any other device on the bus which then assumes total authority over all 'conversations'. To do this, the IEEE488 system uses five management lines and three control lines, in addition to the eight data lines.

Fortunately, there is only one 'talker' – the BBC machine, and one 'listener' – the printer, so we only need use one of the management

Program 1. Note the slight difference between 1 and l in printing, for example in lines 10 and 170. Assume 0 is O, apart from words such as PROC, MOD. Also _ will appear on screen as –, [as ←, and] as →. This is because of the teletext character set, and will not affect the program.

```
 10 N=0:PROCassemble
 20 N=2:PROCassemble
 30 PRINT"Initialise at &";~init
 40 PRINT"Byteout routine &";~byteout
 50 PRINT"Restore to normal &";~restore
 60 PRINT" *SAVE";CHR$34;"PET";CHR$34~init" ";~end+23;" ";~init
 70 CALLinit
 80 END
 90
100 DEFPROCassemble
110 IF N=0 table=&D00
120 T=table
130 !T=&5020100:T!4=&9010000:T!8=&508:T!12=&2000404
140 byte=T+16
150 printon=T+17
160 lcase=T+18
170 vdulflag=T+19
180 bytecount=T+20
190 oldvec=T+21
200 wrchvec=&20E
210 PB=&FE60
220 PA=PB+1
230
240 P%=&C00
250 [OPT N
260 .init
270 LDA wrchvec
280 STA oldvec
290 LDA wrchvec+1
300 STA oldvec+1
310 LDA#start MOD 256
320 STA wrchvec
330 LDA#start/256
340 STA wrchvec+1
350 LDA#3
360 STA PB+2
370 LDA#0
380 STA printon
390 STA vdulflag
400 STA lcase
410 STA bytecount
420 LDA#&24
430 LDX#1
440 JSRbyteout
450 LDA#13
460 LDX#3
470 JSRbyteout
480 RTS
490
500 .restore
510 LDA oldvec
520 STA wrchvec
530 LDA oldvec+1
540 STA wrchvec+1
550 RTS
```

page 38▶

lines. But we do have to use all three control lines as well as the data lines. This is why we have to make use of both ports of the 6522 VIA. Port A, which is designated as the printer port, is used to output the data, and parts of port B, the user port, for control and management. PB0 and PB1 are used as outputs and PB6 and PB7 as inputs.

According to the *User Guide* (page 408) it is simplicity itself to write and install your own machine code printer routines. 'The address of the user routine is placed at location &222 and the user defined routine can be selected with *FX 5,3.' Unfortunately, this turns out to be nowhere near as simple as it sounds. It requires a much deeper knowledge of the operating system workings than is generally available

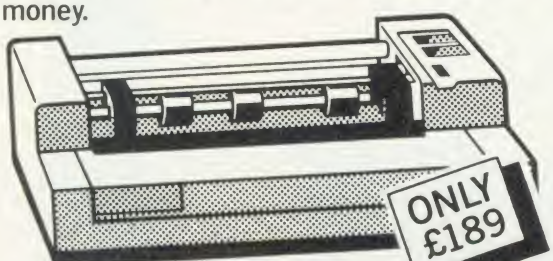
page 41 ▶

▶ page 37. Program 1 continued

```
560
570 .byteout
580 STX PB
590 EOR #&FF
600 STA PA
610 .NRFD
620 BIT PB
630 BPL NRFD
640 DEX
650 STX PB
660 .NDAC
670 BIT PB
680 BVC NDAC
690 INX
700 STX PB
710 RTS
720
730 .start
740 STA byte
750 TXA:PHA
760 TYA:PHA
770 LDY#0
780 LDX#131
790 JSRcharcheck
800 PLA:TAY
810 PLA:TAX
820 LDA byte
830 JMP(oldvec)
840
850 .charcheck
860 BIT vdulflag
870 BPL countdown
880 STY vdulflag
890 LDA byte
900 JSR byteout
910 JMP returnnull
920
930 .countdown
940 LDA#&FF
950 BIT bytecount
960 BEQ ctrlcheck
970 DEC bytecount
980 RTS
990
1000 .ctrlcheck
1010 LDA byte
1020 AND #&E0
1030 BEQ isitl
1040 LDA byte
1050 JSR outchar
1060 RTS
1070
1080 .isitl
1090 LDA byte
1100 CMP#1
1110 BNE isit2
1120 STX vdulflag
1130
1140 .returnnull
1150 STY byte
1160 RTS
1170
1180 .isit2
1190 CMP #2
1200 BNE isit3
1210 STX printon
1220 JMP returnnull
1230
1240 .isit3
1250 CMP#3
1260 BNE isitl3
1270 STY printon
1280 JMP returnnull
1290
1300 .isitl3
1310 CMP#13
1320 BNE lessthan16
1330 JSR outchar.
1340 RTS
1350
1360 .lessthan16
1370 AND#16
1380 BNE setcounter
1390 RTS
1400
1410 .setcounter
1420 LDA byte
1430 AND#15
1440 TAX
1450 LDA table,X
1460 STA bytecount
1470 RTS
1480
1490 .outchar
1500 BIT printon
1510 BMI CRcheck
1520 RTS
1530
1540 .CRcheck
1550 CMP#13
1560 BNE casecheck
1570 STY lcase
1580 JSR byteout
1590 RTS
1600
1610 .casecheck
1620 AND#&60
1630 CMP#&60
1640 BNE ucase
1650 BIT lcase
1660 BMI lcaseout
1670 LDA#17
1680 JSR byteout
1690 STX lcase
1700
1710 .lcaseout
1720 LDA byte
1730 AND#&DF
1740 JSR byteout
1750 RTS
1760
1770 .ucase
1780 BIT lcase
1790 BPL ucaseout
1800 LDA#145
1810 JSR byteout
1820 STY lcase
1830
1840 .ucaseout
1850 LDA byte
1860 JSR byteout
1870 LDA byte
1880 CMP#34
1890 BNE end
1900 JSR byteout
1910 .end
1920 RTS
1930
1940 .table
1950 ]
1960 ENDPROC
```

Follow us to MICROAGE ELECTRONICS for printer value

At Microage you can now get the very best value in microcomputer printers. We've selected only the best printers available so you can choose confidently, knowing you're getting the best value for money.



The Incredible AP80

The AP80 is probably the world's LOWEST COST, compact 80-column graphic dot-matrix printer available.

It can produce single and double width characters and has the ability to produce any pattern through its dot-matrix capability.

The AP80's robust construction and its unique "unihammer" make it an extremely cost effective and efficient printer.

Its features include

80 Cols 30 cps ● Dot Matrix Unihammer action ● 96 ASCII standard characters ● Up to 8" paper width ● Upper and lower case ● Double width printing ● Standard interface: Centronics

The Amazing AP100

Big brother to the AP80 the AP100 is a wider more sophisticated version of the AP80 with a full graphics capability and the ability to take standard width computer stationery, the AP100 is an ideal choice for anyone with a microcomputer.

Its features include:

80 cols 30 cps ● Dot Matrix Unihammer action ● 116 ASCII standard characters ● Full graphics capability ● Up to 10" paper width ● Upper and lower case ● Double width printing ● Centronics interface: ● Friction feed. Only £215.

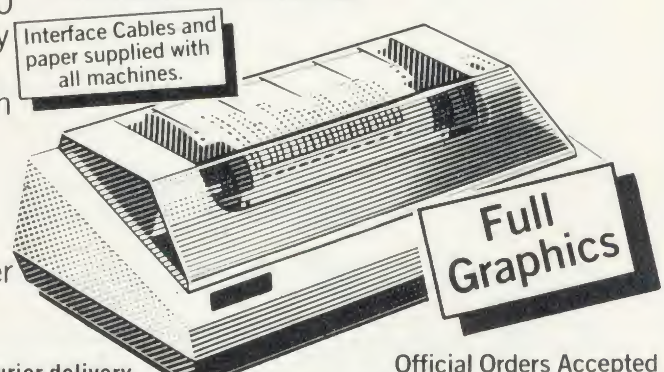
The Outstanding MX80 FT Type 3

The MX80 is a flexible, high quality and extremely reliable graphic, dot matrix printer.

Its superb performance is due to a number of outstanding features. High resolution graphics, a wide range of type sizes, a quality 9 x 9 dot matrix head and up to 132 chars per line are just some of the exceptional features on this ideal printer for microcomputers. Only £390.

Specifications

132 cols 80 ● Logic-seeking, Bi-directional printing ● 9 x 9 Dot matrix printhead ● 96 ASCII standard chars with descenders ● Full graphics capability ● Several type sizes ● Automatic underlining ● Standard interface Centronics ●



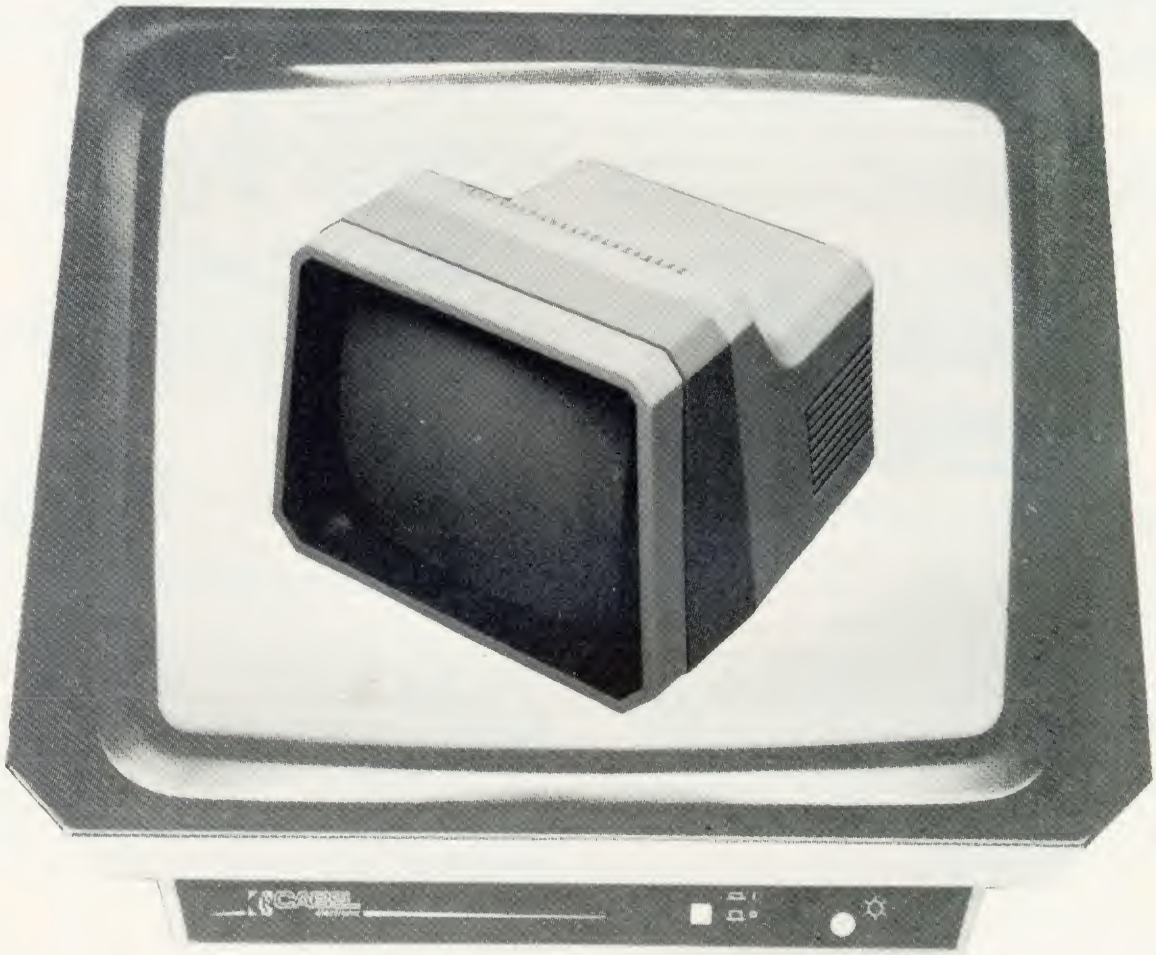
Free courier delivery

Official Orders Accepted

MICROAGE ELECTRONICS

135 HALE LANE EDGWARE MIDDLESEX HA8 9QP TEL: 01-959 7119 TELEX 881 3241

DEALER ENQUIRIES WELCOME CALL 01-959 7119 FOR TRADE PRICES


CABEL
electronic

▶ *page 38*

at present. Back to the drawing board!

The other alternative given by the *User Guide* is to intercept the operating system write character (OSWRCH) routine. The problem with this is that all the VDU codes are sent using OSWRCH. For example, DRAW 10,10 sends six bytes through OSWRCH to the VDU drivers. In this case they would be 25,5,10,0,10,0. So if you do intercept OSWRCH, you have to detect each of the control codes and then find out how many of the succeeding bytes to ignore.

The first method is a much neater approach to the problem, but as the information was not easily available I decided to go for the second one, that of intercepting OSWRCH. Every time you send a byte to the screen and/or the printer, the operating system 'jumps indirect' via &20E. In other words, it looks at memory locations &20E and &20F to find the starting address of the printer and VDU routines. &20E is called the WRite CHaracter Vector (WRCHV) because it points to the relevant routines which would usually be in the operating system ROM. The idea then is to extract this number and replace it with the start address of your printer routine which would be stored in RAM. At the end of your routine, you have to jump back to the address originally specified by WRCHV.

Program 1 gives the routines needed to drive the Pet printer,and splits up into six sections. As the listing stands, the routines are assembled from &C00 upwards so that it will fit in below the normal start of Basic programs at &E00. However, to use programmable graphics characters in programs you will have to put the routines elsewhere, since &C00 to &CFF are used to store the new character definitions. Set PAGE = &F00 before you load the program, and change line 240 to P% = &D00. Again, if you have a disc system you will have to find suitable space for the routines which occupy about a page and a half.

Initialisation (lines 260 – 480). To set up the routines initially, or after a BREAK, you have to "CALL &C00" (or wherever you have assembled the routines to). When you do so the value of WRCHV is replaced by the value of the start of the main part of the printer routines, the values of various flags are set to zero, the 6522 VIA is initialised and the Pet printer is addressed. That is to say, there is a particular control sequence required to 'wake up' the printer and define it as a 'listener'. When the printer receives this message, it responds by switching on its 'ready' light. The routine also sends out a carriage return during the initialisation sequence which gives audible confirmation the printer is listening, and prints out anything left in its buffer.

Restoration to normal printer routines (lines 500 – 550). This routine can be called from the keyboard or from within a program to replace the original value of WRCHV in case other types of printers are required. From the keyboard, pressing the BREAK key also resets WRCHV, so you may prefer to leave this routine out.

Byte output routine (lines 570 – 710). This takes the byte which is in the accumulator and executes the necessary 'hand-shaking' to transfer it to the printer. NRFD and NDAC refer to the various control signals on the IEEE488 bus which could be investigated in more detail by studying Fisher and Jensen. (*The Pet and the IEEE488 Bus,* E. Fisher and C.W.Jensen.)

Save and restore the registers (lines 730 – 830). The A, X and Y registers are saved and then restored, after the character has been dealt with, before jumping back into the normal OSWRCH routines.

Dealing with the control codes (lines 850 – 1470). This is summarised in the form of the flow chart in figure 1. If a VDU 1 has just been executed, the current byte is sent to the printer and a null returned to OSWRCH. If a control code requiring a number of data bytes has just been received, the value of bytecount will not be zero, so the byte will simply be passed on to the OSWRCH routines without any further action. Control codes 1, 2, 3 and 13 all have to be dealt with in special ways, and for the rest, the routine looks up, in a table, how many of the succeeding bytes to

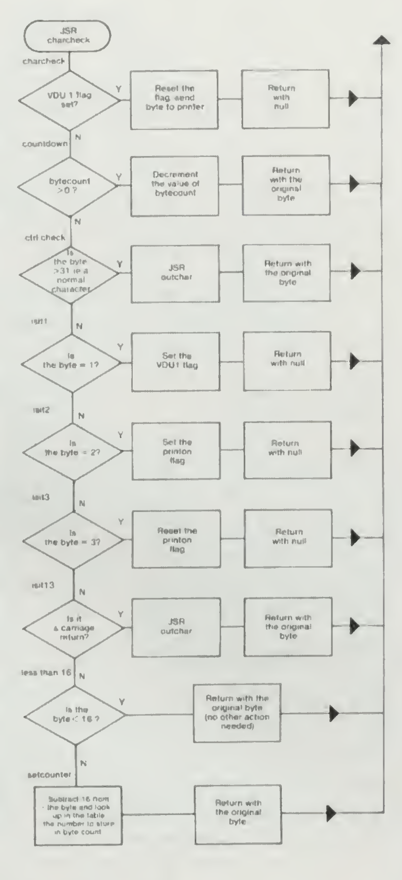


Figure 1.
Checking the bytes to see if they should be sent to the printer

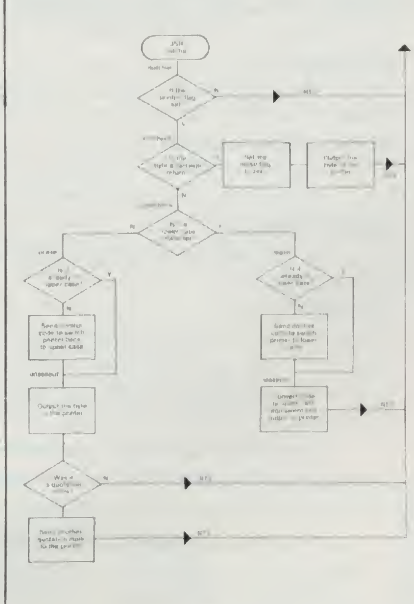


Figure 2.
Outputting a byte to the printer

ignore.

After dealing with 1, 2, and 3, a null is returned to OSWRCH to avoid printer routines in the machine operating system being activated. The VDU 1 call has been implemented as it is on the 0.1 operating system, that is, it is not necessary to have executed a VDU 2 before you try to use VDU 1 to send a character to the printer. If you prefer VDU 1 to work as on the 1.2 operating system, then simply add:

```
883 BIT printon
886 BPL returnnull
```

Dealing with the Pet character set (lines 1490 – 1920). Because the Pet does not use a standard ASCII characer set, there are two problems to be sorted out. (Figure 2 shows the flow chart.) The first problem is that upper and lower case characters are both produced using the same codes (65 – 90). These would normally produce upper case characters, so to make them produce lower case, you have to send out a control code (17). So if a lower case code is sent to the routine you have to send out the control code, convert the lower case code into its upper case equivalent and send that to the printer. To switch back to upper case you have to send out a different control code: 145 which is 17 plus 128. Rather than do this for every lower case character, there is a flag (lcase) which is set when a lower case character is encountered and is checked if successive bytes are in lower case. No further control codes have to be sent out until a character appears which is in upper case. However, after every carriage return, the printer automatically reverts to upper case, so if your text is still lower case, you have to send out another 17 at the beginning of the line. The routines do this automatically.

The second problem is with quotation marks. When an odd number of quotation marks has been sent to the printer, all control codes, rather than being acted on, are printed out as reverse field ASCII characters. If you try to mix upper and lower case characters inbetween quotation marks, they will all appear in upper case interspersed with inverse characters. To overcome this, lines 1870 to 1910 have been added which check whether the output character is a quotation mark and if so to send it out to the printer a second time. Every time a quotation mark is requested, two are sent.

This tends to look a bit untidy, because every line that has one or more quotation marks in it sticks out beyond all the others. Despite this drawback, I feel this is the lesser of the two evils. If you are only using the printer for listings, you may omit this feature by leaving out lines 1870 – 1910.

One other problem is that if programs have lines longer than vthe 80 characters which the printer has on each line, instead of folding over onto the next line, the printer may swallow the whole line and print just the carriage return. I could find no logic in this so avoid long lines.

Figure 3 gives the connections

in the IEEE connector and the lines on the 6522 VIA, on the BBC microcomputer, to which they should be connected. To make use of the existing Pet computer to printer lead, you will have to make a double-sided edge connector, so that it looks like the edge of the printed circuit board (PCB) at the back of the Pet. One possible way of doing this is to buy an old scrap double-sided PCB that has a suitable edge connector. Fix it into a small plastic box and wire it up from there to the BBC's own connectors using a 26-way jumper connector to the printer port and a 20-way jumper to the user port. The connections on the edge connector in fact copy the positions on the IEEE connector (figure 4).

Alternatively, try to get hold of the proper, but rare,IEEE Amphenol connector and wire it up to the jumper leads.

Individual control codes can be sent to the printer using VDU 1. For example, VDU 1, 18 outputs an 18 to the printer which has the effect of causing all the remaining characters on the current line to be printed in reverse field (white on black), unless a VDU 1,146 is executed which would cause it to revert to normal printing. The full set of characters, including the Pet graphics characters which can be produced by a combination of VDU 1,17 and VDU 1,18 can be printed out by running program 2.

The effect of VDU 1,1 is that succeeding characters on the line will be enhanced, ie instead of being a matrix of seven dots by six dots, the width will be increased to 12 dots. If you want to revert to normal printing on any one line, you have to send out VDU 1, 129 (129 = 128 + 1). This effect is actually cumulative in that if you use VDU 1,1 twice on the same line, all characters following the second call will be 18 dots wide. Another VDU 1,1 would make them 24 dots wide and so on. But as with VDU 1,17 and VDU 1,18, their effect is removed by a carriage return, so that at the beginning of every line you are in unenhanced, non-reversed upper case.

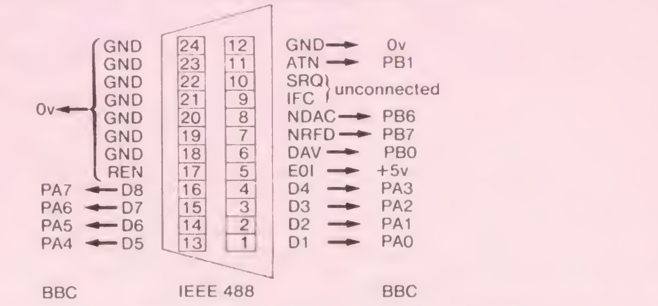


Figure 3. Interconnections between IEEE488 connector and BBC machine

The Commodore printer is actually a very intelligent beast. It has its own processor and can perform various tricks. One is to produce a user-defined graphics character and another to format the data sent to the printer. To see how to access these special functions, it is necessary to look at what happens when you address the printer.

When you first want the printer to become a 'listener', you warn all devices on the bus that an address is coming by making the ATN line low. (IEEE488 is all negative logic, so to make devices pay attention, make the attention line low. Also note that in the byte – out routine the byte is inverted (EOR#&FF) before being put onto the data lines.) The printer's device number is four, so to tell it to listen, you have to send out &24. ATN is then put back to logic 1, and any further data put on the lines is printed out, or acted on in the case of control codes.

The various special facilities which the printer is capable of are accessed using a secondary address. Having sent out the primary address, &24, you keep the ATN line low and send out a secondary address. Program 3 gives an example of this, and shows how to define and use the programmable graphics character. The secondary address in this case is &65, and after sending it out, you take the ATN line high and send out the six bytes of data used to define the character. To show that you have finished with the secondary address feature, send out a &3F – the universal un-listen code! After that, you start again with the normal primary addressing procedure and carry on printing.

There are other tricks using different secondary addresses,so consult the manual.

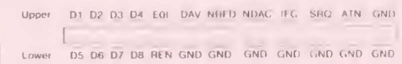


Figure 4. Pet end connector

*This program was developed for the 3022 series. The only modification for other series is to allow a line feed character with each carriage return. This is done by changing line 1260 to BNE isit10 and adding:

```
1291 .isit10
1292 CMP#10
1293 BNE isit13
1294 JSR outchar
1295 RTS
```

```
 10 @%=5
 20 PRINTCHR$2'"  DEC  HEX  NML VDU17 VDU18 VDU17,18      ";
 30 PRINT"DEC  HEX  NML VDU17 VDU18 VDU17,18"'
 40 FOR J%=32 TO 95
 50   Q%=J%+128
 60   PROCset(J%)
 70   PRINT SPC(7);
 80   PROCset(Q%)
 90   PRINT
100   NEXT
110 VDU3
120 END
130
140 DEFPROCset(N%)
150 PRINT N%;"   "~N%;"   ";
160 PROCchar
170 VDU1,17:PROCchar
180 VDU1,18:PROCchar
190 VDU1,17,1,18:PROCchar
200 ENDPROC
210
220 DEFPROCchar
230 PRINTCHR$(N%);
240 VDU1,145,1,146
250 IF N%=34PRINT"   "; ELSE PRINT"   ";
260 ENDPROC
```

Program 2.
Outputs the printer's complete character set

```
 10 byteout=&C45
 20 X%=1:A%=&3F:CALL byteout
 30 A%=&24:CALL byteout
 40 A%=&65:CALL byteout
 50 VDU1,20,1,54,1,118,1,118,1,55,1,20
 60 X%=1:A%=&3F:CALL byteout
 70 A%=&24:CALL byteout
 80
 90 VDU2
100 PRINT'"Acorns Rule UK?"'
110 FOR N%=1 TO 40
120   VDU1,254,1,32
130   NEXT
140 VDU13,3
150 END
```

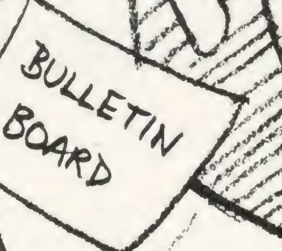
Program 3.
Example of using the programmable printer character

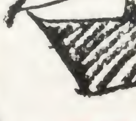
MORE IDEAS THA
MORE PROGRAMS THA
LESS MONEY THA

PROGRAMS FOR EDUCATION, HOME, BUSINESS, AND GAMES – FOR SINCLAIR, BBC, RESEARCH MACHINES, APPLE, COMMODORE, ACT SIRIUS, TRS-80, IBM, ICL, and more each month. . .

HOW THE MICRONET 800 SERVICE WORKS:

Most micros connect by a standard DIN plug and coaxial cable to an adaptor. They're portable, so you can use the system wherever you put your computer and telephone.





Simply dial Micronet's number and enter your subscriber number into the keyboard. A 'welcome to Micronet' message appears on the screen.
You're on-line to one of the world's largest databases.

PRESTEL
MICRONET 800's
thousands of ideas — packed pages – games, education, business software, club news and information.

When you have all the information you need – or you've finished 'downloading' software to your computer's memory – just switch off the telephone link at your keyboard and replace the receiver.

BURIED TREASURE

The problems in the September issue of *Acorn User* connected with ciphers generated an enormous response, with many people saying how much they liked trying to solve them. Since then, of course, we've had the 'Soviet Spy in Government Communications Headquarters Scandal' – and the sentencing of one Geoffrey Prime to 38 years imprisonment for giving the Russians details of Nato ciphers and Warsaw Pact codes which the West had broken.

The Government is not willing to release anything other than the barest details, though why, since the Russians presumably know them all, they should be kept from us is beyond me. I have therefore written to Mrs Thatcher, drawing her attention to the tremendous cipher-cracking talents of the readers of *Acorn User* and suggested that in future we should publish overheard Warsaw Pact transmissions in these pages to see if our readers can help out GCHQ Cheltenham!

The September issue featured a straightforward substitution cipher and a message based on a single keyword where each letter of the keyword was added in rotation to the letter in the message being encoded, assuming for the sake of simplicity that A follows Z.

A more secure way of encrypting messages is a book cipher. Here each letter of a piece of prose known to both the sender and the recipient is numbered consecutively. Each letter of the plain text message is then assigned a number from the prose passage which corresponds to the same letter.

Suppose, for example, we wish to despatch the message 'SEND GOODS HERE' and the cipher we are using is based on the book of *John*, ("In the beginning was the Word and the Word was with God . . ."), one way to do it is:

17 30 26 24 8 32 43 34 37 29 7 23 20

This is clearly safer than a keyword cipher since repeated letters or numbers – so often the entrypoint for a good cryptanalyst – occur infrequently, though as with all ciphers of this nature, if the enemy gets enough messages he will crack the code sooner rather than later.

The most glamorous – and frustrating – example of a book cipher is the Beale Treasure Cipher, which was with us long before metal detectors and the bestseller *Masquerade* encouraged everyone to go digging up the English countryside.

In 1885 one James Ward of Virginia published a pamphlet in which he described how an explorer named Thomas Jefferson Beale had buried a fantastic treasure 60 years earlier and had left his legacy to posterity in three separate ciphers. The first purported to describe the precise location of the vault in Bedford County, Virginia which contained the loot.

Cipher 2 described the contents of the vault in fairly mouthwatering terms, while the third described whom Beale thought should benefit from all this wealth – and such is

Beale's Treasure Cipher – does a fortune still await the first person to crack his code?

71 194 38 1701 89 76 11 83 1629 48 94 63 132 16 111 95 84 341 975
14 40 64 27 81 139 213 63 90 1120 8 15 3 126 2018 40 74 758 485
604 230 436 664 582 150 251 284 308 231 124 211 486 225 401 370
11 101 305 139 189 17 33 88 208 193 145 1 94 73 416 918 263 28
500 538 356 117 136 219 27 176 130 10 460 25 485 18 436 65 84 200
283 118 320 138 36 416 280 15 71 224 961 44 16 401 39 88 61 304
12 21 24 283 134 92 63 246 486 682 7 219 184 360 780 18 64 463
474 131 160 79 73 440 95 18 64 581 34 69 128 367 460 17 81 12 103
820 62 116 97 103 862 70 60 1317 471 540 208 121 890 346 36 150
59 568 614 13 120 63 219 812 2160 1780 99 35 18 21 136 872 15 28
170 88 4 30 44 112 18 147 436 195 320 37 122 113 6 140 8 120 305
42 58 461 44 106 301 13 408 680 93 86 116 530 82 568 9 102 38 416
89 71 216 728 965 818 2 38 121 195 14 326 148 234 18 55 131 234
361 824 5 81 623 48 961 19 26 33 10 1101 365 92 88 181 275 346
201 206 86 36 219 320 829 840 68 326 19 48 122 85 216 284 919 861
326 985 233 64 68 232 431 960 50 29 81 216 321 603 14 612 81 360
36 51 62 194 78 60 200 314 676 112 4 28 18 61 136 247 819 921
1060 464 895 10 6 66 119 38 41 49 602 423 962 302 294 875 78 14
23 111 109 62 31 501 823 216 280 34 24 150 1000 162 286 19 21 17
340 19 242 31 86 234 140 607 115 33 191 67 104 86 52 88 16 80 121
67 95 122 216 548 96 11 201 77 364 218 65 667 890 236 154 211 10
98 34 119 56 216 119 71 218 1164 1496 1817 51 39 210 36 3 19 540
232 22 141 617 84 290 80 46 207 411 150 29 38 46 172 85 194 36
261 543 897 624 18 212 416 127 931 19 4 63 96 12 101 418 16 140
230 460 538 19 27 88 612 1431 90 716 275 74 83 11 426 89 72 84
1300 1706 814 221 132 40 102 34 858 975 1101 84 16 79 23 16 81
122 324 403 912 227 936 447 55 86 34 43 212 107 96 314 264 1065
323 328 601 203 124 95 216 814 2906 654 820 2 301 112 176 213 71
87 96 202 35 10 2 41 17 84 221 736 820 214 11 60 760

the state of human nature that few people seem to have troubled themselves with it!

Ward himself claimed to have translated the second . It was a book cipher created by numbering the initial letters of each of the 1322 words of the American Declaration of Independence. Each plain text letter was then replaced with the number of a word beginning with that particular letter.

Cipher 2, when decoded, ends with the intriguing statement, 'Paper number one describes the exact locality of the vault so that no difficulty will be had in finding it.'

Cipher 1 has proved a tougher nut. It consists of 495 numbers from 1 to 2096 and has so far defied the combined attacks of several generations of amateur and professional cryptanalysts. The fact that the numbers extend to 2096 would, on the surface, seem to exclude the use again of the Declaration of Independence – or at least the use of only the initial letters of the words, as in cipher 2.

However, an artificial intelligence expert has published a paper which carries a statistical analysis of cipher 1 and suggests that it *was* based on the Declaration, but that the whole thing was a hoax*. Still, it's very difficult to *prove* a document is meaningless and many people still believe a fortune awaits whoever cracks the key. Certainly if it was an elaborate jape, the author seems to have taken his secret to the grave with him. Those with a taste for these things can see if their microcomputer can go where no cryptanalyst hs so far managed to tread.

*James T. Gillogly in *Cryptologia*, vol 4, no 2, April 1980. I am indebted to Mr Gillogly for supplying much of the background information to the Beale Cipher.

This month you're an eager-beaver cipher clerk monitoring enemy signals. A regular transmission comes through which you know to be a simple substitution code. The message for today is:

```
SNDRR PLGPS DSHNW PWIZP HKWDR PPIQS HNWQT HICHP DMHKT SYYYY
TIPDF PZIDM IRDXF ZHSTP IQLMH SFDWU ODBIS NNODW FOHCS DBPII
IIOHP SDBIP WIZQH VTIMM IVRDQ IPIYH PSHWK VMNQI ULMID WUPSD
MSPSZ IRAIT LWUMI UTNLM PSNON MMNZG DPIUN WDCDO NLPVN IOIAI
MFUDF HTDSI VNISM F
```

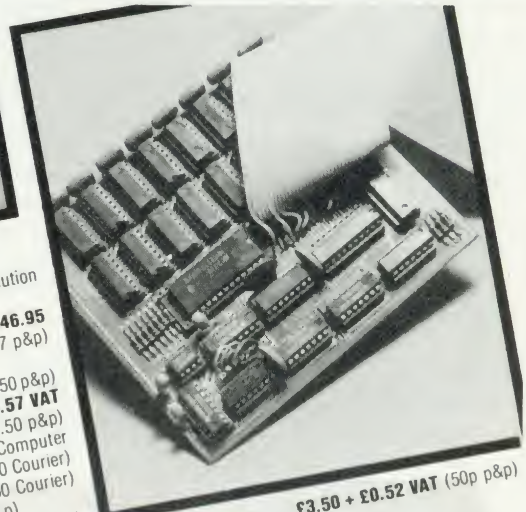
The following day another message comes through:

```
28  11  66  53  53  52  73  222  52  10  26  61  29  39  42
52  49  44  72  70  17  35  62  29  91  15  66  125  45  52
34  39  73  26  91  38  39  30  31  61  23  86  3  163  93
70  62  69  85  66  52  49  10  61  58  54  125  76  59  35
54  59  30  163  39  32  37  61  44  8  25  60  54  222  53
3  76  29  52  62  39  40  76  73  163  75  83  72  39  32
163  39  5  56  86  33  40  39  54  86  163  36  73  163  125
163  44  73  163  125  29  7  77  83  39  72  51  163  62
73  163  18  163  54  73  71  86  163  36  73  163  125  83
39  54  32
```

'Good grief! What's the question?' Well, do your deciphering and all will be revealed.

Answers, on a postcard please, to: December Competition, Acorn User, 53 Bedford Square, London WC1B 3DZ. Entries must arrive by February 2. Acornsoft packages worth up to £20 await the three lucky winners.

NEW from our BBC micros accessories range



BBC HARDWARE

Model 'A' 16K RAM 32K ROM. Full colour, high resolution graphics ...... **£260 + £39 VAT** (£7 p&p)
Model 'B' 32K RAM 32K ROM. 16 Colour Graphics ... **£346.95 + £52.05 VAT** (£7 p&p)

BBC Disk Interface ...... **£95 + £14.25 VAT**
BBC 'B' Upgrade Kit .... **£113.04 + £16.96 VAT** (£1.50 p&p)
BBC Single Disk Drive 100K bytes **£230.43 + £34.57 VAT** (£2.50 p&p)

14" RGB Colour Monitor (as used in the BBC Computer Programme) .... **£270 + £40.50 VAT** (£9.50 Courier)
12" Green Monitor **£82.60 + £12.40 VAT** (£6.50 Courier)
BBC Joystick ...... **£11.30 + £1.70 VAT** (£1 p&p)
Acorn GP 100 Printer ...... **£200 + £30 VAT** (£4.50 p&p)
Epson MX80 FT111 Printer, new model, dot matrix, high res. graphics, 80 or 132 char's per line ...... **£325 + £48.75 VAT** (£4.50 p&p)

Printer Cable (parallel) ...... **£13.04 + £1.96 VAT** (£1 p&p)
Tapes (Computer standard) ..... **£0.86 + £0.14 VAT** (£1 p&p any quantity)

Disks ('BBC' standard/pack of ten) ... **£19.50 + £2.93 VAT** (£1 p&p)

Books ...... Prices on application
Cover for BBC ...... **£4.99 + £0.75 VAT** (£1 p&p)
Cassette Recorders ...... **£25.22 + £3.78 VAT** (£3 p&p)
Teletext ...... available Jan '83
Econet ...... available Jan '83
Dual Disk Drives 800K ...... available Jan '83

Flochart Stencil ...... **£3.50 + £0.52 VAT** (50p p&p)
Acorn soft ...... List Prices

A to D

A highly accurate and PRECISE convertor for your external devices .... **ONLY £31.50 plus £4.73 VAT** plus £1.50 p&p

FAN

For the professional man to keep the computer COOL .... **Price £33.25 plus £4.99 VAT** plus £1.50 p&p

LIGHT PEN

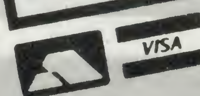
A complete program with plug-in light pen that can be included in your program. Ideal for games and data input ... **ONLY £30 plus £4.50 VAT** plus £1 p&p.

GRAPHICS TABLET

Transfer your technical drawings or designs from pad to screen in seconds and in colour. 22″ × 27″ approx. Measures distances and areas, and moves drawings on screen ... **ONLY £75 plus £11.25 VAT** plus £4.50 p&p.

128K Ram Pack

Simple plug-in memory expansion board, addressable for your own programs .... **Price £299 plus £44.85 VAT** plus £1 p&p.



Mail or telephone order. Barclaycard or Access Credit Card accepted.

MICRO MANAGEMENT

Micro Management, 32 Princes Street, IPSWICH. Telephone 59181

C.J.E. Microcomputers

BBC Specialists

VAT Included where applicable

Quality Disk Drives

Single drive	40 track single sided 1 × 100k	£200.00
Dual drive	40 track single sided 2 × 100k	£350.00
Dual drive	40 track double sided 2 × 200k	£546.25
Dual drive	80 track double sided 2 × 400k	£799.25

All drives are cased with own PSU for reliability, and include connecting cables and utilities disk. Delivery £4.00

Software for the BBC Micro

MISSILE CONTROL the first implementation on the BBC Micro of the popular arcade game. (32K)	£9.00
MAZE MAN an authentic version of the popular arcade game. (32K)	£6.00
BALLOONS a highly original game that soon becomes compulsive playing. (32K)	£6.00
DISSASSEMBLER the memory dump routine includes a scrolling back in memory facility. (16K)	£5.00

MISSILE CONTROL, MAZE MAN and BALLOONS use the Keyboard or Joysticks for control

30 + Programs for the BBC Microcomputer

This Book contains program listings, with explanations and tips on using the BBC Micro
GAMES UTILITIES GRAPHICS and MUSIC
Most programs will run on Model's A and B
Edited by C. J. Evans, various Authors.
A pair of cassettes with all the programs is available.

BOOK £5.00
BOOK and CASSETTE SET £9.00

LEADS

The BBC Micro comes without a cassette lead

7Pin Din to 2 × 3.5mm and 1 × 2.5mm minijacks	£4.00
7Pin Din to 5Pin Din and 2.5mm minijack	£4.00
7Pin Din to 7Pin Din	£4.00
7Pin Din PLUGS	Two for £0.65
6Pin Din PLUGS (for RGB socket)	Two for £0.65
5Pin Din PLUGS (360' for RS232)	Two for £0.65

RS423 TO RS423 (BBC Micro to BBC Micro)
Two metre cable £4.00 Four metre cable £5.00

TELEVISION/MONITOR LEADS full range available

Phono plug to Co-ax with high quality cable 3 Metres	£3.00
BNC Plug to BNC Plug	£3.10
BNC Plug to Phone plug (i.e. BBC Micro to Rediffusion TVRM)	£2.20
RGB 6PIN DIN to 6PIN DIN	1 metre £4.00 2 metre £5.00

PRINTER CABLES

BBC to 36 way Centronics Type connector	£17.50
BBC to 25 way D Type (For use with RS232)	£ 9.50
BBC to 40 way edge connector (Centronics 739)	£20.00
TORCH to 36 way Centronics Type connector	£20.00

Blank C30 Computer Cassettes

Ten for £4.50
15 Way D Type Plug with Cover £2.75
Computer graphics design pads 100 sheets £4.00

BBC Upgrade Kits

RAM UPGRADE (100ns)	£23.00
KIT A Printer and I/O Port	£ 9.50
KIT B Analogue Port	£ 8.00
KIT C Serial I/O and RGB	£10.00
KIT D Expansion Bus/Tube	£ 7.50
Full Upgrade kit	£58.00

All components full specification

STAR DP8480 PRINTER From £250.00 Inc VAT

80 CPS : 80/96/132 COLS
BIDIRECTIONAL LOGIC SEEKING
TRACTOR WITH FRICTION FEED

CENTRONICS £217.39 + £32.61 VAT = £250.00
RS232 £235.00 + £35.25 VAT = £270.25

High Res Graphics option to allow BBC Screen dumps £15.00/£20.00
(24HR SECURICOR DELIVERY FOR PRINTERS £8.00)

VAT Included where applicable

Send SAE for full Price List of our large range of accessories.

C.J.E. Microcomputers

POSTAGE Add 50p per order or as stated

Dept (AU), 25 HENRY AVE, RUSTINGTON, W.SUSSEX. BN16 2PA (09062) 6647

MIDWICH COMPUTER COMPANY LIMITED

FAST EX-STOCK DELIVERY OF MICROCOMPUTER COMPONENTS AT UNBEATABLE PRICES

MIDWICH NOW APPOINTED OFFICIAL ACORN BBC MICRO DEALER

ACORN ATOM UPGRADES

Description	Price
*Atom disc pack (AAH06)	299.00**
*Atom colour card (AAH07)	39.25**
*4K Atom DOS (AAS41)	60.00**
*4K FP ROM (AAS22)	20.00
*BBC ROM Set (XXXXX)	43.43**

BBC Microcomputer Upgrades

*Memory Upgrade (BBC1)	21.50
*Printer & User I/O kit (BBC2)	7.60
*F Disc interface inc DOS (BBC3)	70.00
*Analogue input kit (BBC4)	7.60
*Serial I/O & RGB kit (BBC5)	10.25
*Expansion bus & tube kit (BBC6)	5.95
*Printer cable inc Amphenol plug (not assembled) (BBC21)	13.00
*User port connector & cable (BBC22)	2.00
*Analogue input plug & cover (BBC44)	2.25
*5 pin DIN plug for serialint (BBC111)	0.60
*6 pin DIN plug for RGB int (BBC109)	0.60
*7 pin DIN plug for cassette (BBC141)	0.60
*Connector for bus & cable (BBC66)	3.50
*Single disc drive (100K) (BBC31)	225.00*
*Dual disc drives (BBC32)	345.00*
Teletext Receiver (200K) (BBC71)	144.34**
Prestel receiver (BBC72)	90.00**
*Games Paddles (per pair) (BBC45)	11.30

*Prices on these items are likely to change please contact sales office before ordering
As some items are on extended delivery from Acorn please check availability before ordering items marked **

ACORNSOFT FOR THE ATOM

Description	Price
Word pack ROM inc manual (AC100)	26.00
Atom Forth (AC101)	10.00
Forth Theory & Practice (AC102)	6.00
Forth Utilities (AC103)	10.00
Atom Synthesiser (AC104)	10.00
Atom Database inc manual (AC105)	10.00
Utility pack 1 (Dissassembler Fast cos. Renumber) (AC106)	10.00
Games pack 1 (Asteroids, Sub-hunt, Breakout) (AC107)	10.00
Games pack 2 (Dogfight, Zombie, Mastermind) (AC108)	10.00
Games pack 3 (Rat Trap, Lunar Lander, Black Box) (AC109)	10.00
Games pack 4 (Star trek, Four Row, Space Attack) (AC110)	10.00
Games pack 5 (Invaders Reversi, Wumpus) (AC111)	10.00
Games pack 6 (Dodgems, Simon, Amoeba) (AC112)	10.00
Games pack 7 (Life Forms, Ballistics, Snake) (AC113)	10.00
Games pack 8 (Stargate, Go Moku, Robots) (AC114)	10.00
Games pack 9 (Snapper, Babies, Minotaur) (AC115)	10.00
Games pack 11 (Missile base, Snooker, Dominoes) (AC116)	10.00
Atom Adventures (Dungeon, Intergalactic House) (AC117)	10.00
Atom Chess (AC118)	10.00
Atom Calc (AC119)	34.00
Atom Lisp (AC120)	15.00
Lisp Theory & Practice (AC121)	6.00

ACORNSOFT FOR THE BBC MICROCOMPUTER

Graphs & charts on the BBC Micro (AC122)	7.50
Graphs & Charts Cassette (AC123)	7.50
Algebraic manipulation pk (AC124)	8.65
Forth on BBC Micro (AC125)	7.65
Forth pack (AC126)	14.65
Lisp on BBC Micro (AC127)	7.50
Lisp pack (AC128)	14.65
Games-Philosophers quest (AC129)	8.65
Games-Defender (AC130)	8.65
Games-Monster (AC131)	8.65
Games-Snapper (AC132)	8.65

NEW EPROM PROGRAMMER FOR BBC MICROCOMPUTER

★ Programs 2516, 2716, 2532, 2732 Industry Standard EPROMS
★ No external power supply required
★ Plugs straight into expansion socket
★ Easy to use
★ Includes all Software required

EPROM Programmer (Kit)	49.95
EPROM PROGRAMMER (Assembled)	57.95

ACORN & BBC MICRO COMPONENTS

2114 Low Power 200ns	0.80
4516/4816 100ns	2.69
6522	2.99
74LS244	0.59
74LS245	0.69
81LS97	0.90
DP8304	4.50
DS3691N	4.50
DS88LS120N	4.50
20 way Right Angle IDC Header	2.46
26 way Right Angle IDC Header	3.24
34 way Right Angle IDC Header	3.80
15 way Right Angle D. SKT	3.50

VISA

24 Hour Telephone order service for credit card holders.
All prices exclude VAT and carriage (0.75 on orders under £10 nett)
Official orders from educational and government establishments, and public companies accepted. Credit accounts available to others (subject to status)
All orders despatched on day of receipt. Out of stock items will follow on automatically at our discretion or a refund will be given if requested.
NO SURCHARGE FOR CREDIT CARD ORDERS

MIDWICH COMPUTER CO LTD

Rickinghall House, Hinderclay Road, Rickinghall, Suffolk IP22 1HH
Telephone Diss (0379) 898751

ATOM TO BEEB

As a teacher, C.J. Hollyman finds little time to sit down with a BBC micro at work, but some spare time at home with his Atom. This article describes how he writes programs on his Atom to read into his Beeb later.

Do you have limited access to a BBC computer and yet want to write programs for it? If you have an Acorn Atom your problem is solved. This program is designed so BBC programs can be written on the Atom, stored on tape, and later read into the Beeb. However, the program does not convert from Atom Basic into BBC Basic. Programs must be entered in BBC Basic and so (in general) will not run on the Atom.

The listing shown takes a program written in BBC Basic, starting at memory location 2900 hex (the usual start address of a program in the lower text space of the Atom) and writes it as a series of ASCII files each 256 bytes long complete with a header starting at location 9600 hex. Each file is written in such a way that when it is stored on tape, the BBC micro can read it.

Subroutine (a) writes the header which is the same for all blocks, so the data is recorded as a series of single blocks. The counter Z, informs the user how many blocks there are. If the BBC program is not entered into the Atom at the usual place in memory, the value of D in line 2060 must be changed.

Subroutine (b) writes the machine code routine which is used in recording the file. If another area of memory instead of 9600 hex is used for storing the file, the values in lines 2000 (A), 2040 (J), 2060 (calculating ?#80), 3010 and 3020 (addresses in the machine code routine), must be changed.

Subroutine (c) writes the BBC program as an ASCII file. Lines 4010 and 4020 convert the line numbers (F) to ASCII values, and lines 4030 and 4040 copy the rest of the line. Lines 4040 to 4080 check if the program is completely copied or if there is not enough memory space to write the next line and fills the rest of the file with a dummy line – line 30000 which is a REM statement.

Subroutine (d) uses subroutine (f) to calculate the cyclic redundancy check, and subroutine (e) writes the file to tape, giving the usual prompts.

There are two parts to using the program, the first of which is writing tapes. Enter the BBC program into the Atom via the keyboard. Load BBCWRITE by:

```
?18=#82
NEW
LOAD"BBCWRITE"
```

then run and follow the instructions.

For reading tapes, set the tape read speed of the BBC micro to 300 baud with:

```
*TAPE3
```

then load the program by entering:

```
EXEC"PROGRAM NAME"
```

This last command must be entered for each block, but can be simplified by first entering

```
*KEY 0 *EXEC"PROGRAM NAME"
IM
```

Each block can then be loaded by pressing function key 0. Finally, the dummy line 30000 should be deleted from the program.

page 51

```
232REM BBCWRITE
1010REM C.J.HOLLYMAN. 1982.
1020GOS.a;REM CONSTANTS
1030GOS.b;REM MACHINE CODE
1040DO E=0
1050GOS.c;REM WRITE FILE
1060GOS.d;REM CALCULATE CRC
1070GOS.e;REM TAPE
1080U.E;E.
1090**********************
2000a?#9600=#2A;A=#9601
2010IN.''''"FILENAME"$A
2020A=A+LENA
2030!A=#FF0D0000;A!4=#FF0D00FF;A!8=#FF;A!12=#8001;A!16=0;A=A+17
2040C=0;F.J=#9601 TO A;N=?J;GOS.f;N.
2050A?1=C/256;A?2=C%256;G=A+3
2060?#80=G-#95FE;D=#2901;Z=0;@=1;R.
2070****************
3000bDIMLL(2);F.J=1TO2;DIMP(-1);[
3010:LL0LDX@0;:LL1LDA#9600,X;JSR#FFD1;INX;BNELL1
3020:LL2LDA#9700,X;JSR#FFD1;INX;CPX#80;BNELL2;RTS;]
3030N.;R.
3040*********************
4000cB=G
4010DOF=256*(?D)+D?1;T=10000
4020F.J=1TO5;?B=48+F/T;B=B+1;F=F%T;T=T/10;N.
4030$B=$(D+2);B=B+1+LENB;D=D+1+LEND
4040U.(B+7+LEND)>G+255 OR ?D=255
4050IF?D=255;E=1
4060!B=#30303033;B!4=#4D455230;B=B+8
4070F.J=B TO (G+254);?J=80;N.
4080G?255=13
4090R.
4100********************
5000dC=0
5010F.J=0 TO 255
5020N=G?J;GOS.f
5030N.
5040B=G+256;?B=C/256;B?1=C%256
5050R.
5060*****************
6000eP."RECORD TAPE";LINK#FFE3
6010P.';?#B002=?#B002¦2
6020F.J=1TO300;WAIT;N.
6030?#B002=?#B002&253
6040LINK LL0
6050Z=Z+1;P."BLOCK "Z" SAVED."'
6060P."TURN OFF TAPE."'
6070R.
6080*****************
7000fREM CALCULATE C.R.C. ON NUMBER N
7010REM CALL IT C
7020REM SET C=0 BEFORE START.
7030H=C/256;L=C%256
7040H=N:H;C=256*H+L
7050F.X=1TO8
7060T=0;H=C/256
7070IFH>127;C=C:#810;T=1
7080C=(C*2+T)&#FFFF
7090N.;R.
7100*********************
9999END
```

Extra memory

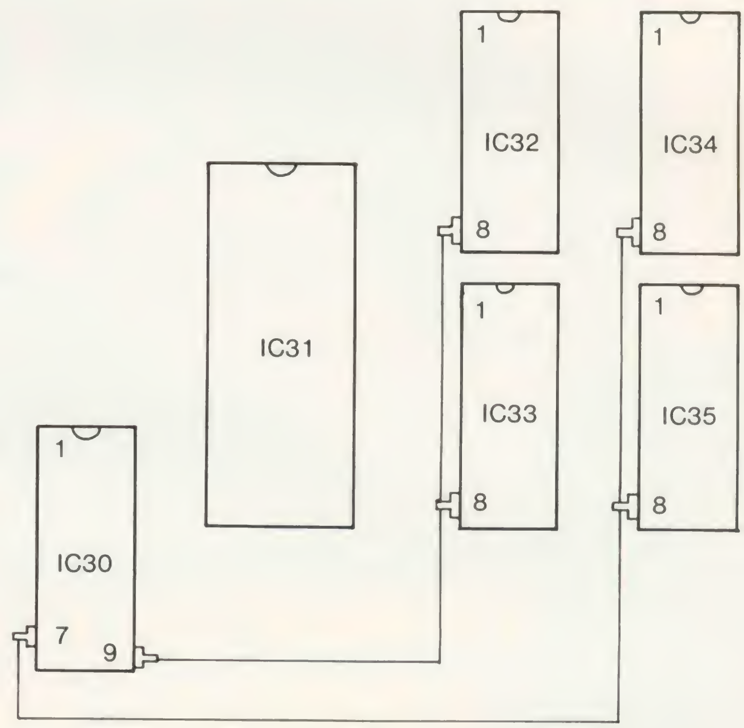
J. Charlton finds 2k hidden away in an expanded Atom

The memory map in the Atom handbook (page 196) shows 2k of unused space (HEX9800 to HEX9FFF). Furthermore, there is no planned use for this space in any of the expansion options.

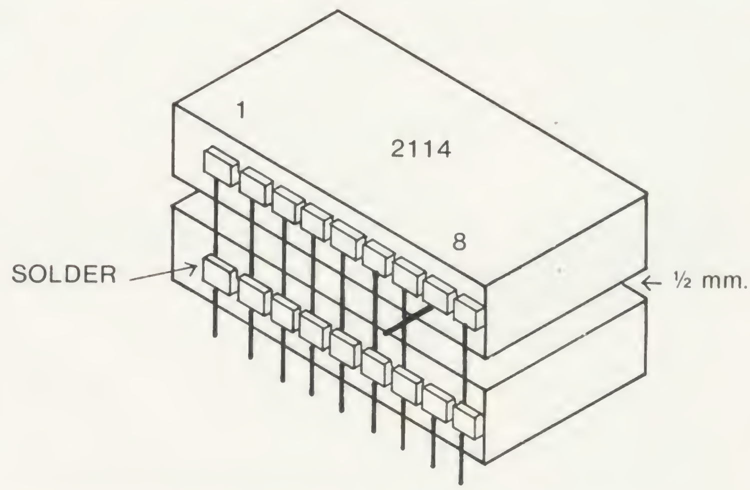
Inspection of the circuit diagram shows that decoded chip enable pulses are available on pins 7 and 9 of IC3. To create additional memory space follow the steps below:

- Obtain four new 2114 N–L chips.
- Bend pin 8 out at 90° on each of these chips, just below the point where the pin thins down.
- Remove IC32, IC33, IC34 and IC35 with care and mount the new chips piggy-back, pin for pin on top of the ones removed. Leave a 0.5mm space for cooling between the chips in each pair.
- Solder all pins except pin 8, take care not to get solder on the lower parts of the pins of the lower chips.
- Replace the combination pairs in sockets 32, 33, 34 and 35.
- Connect the bent out pins of the top chips in position 32 and 33, with thin insulated wire, to each other and to pin 9 of IC3. Repeat this with the top chips in positions 34 and 35, connecting this time to pin 7 of IC3. The final circuit is shown above.
- Test the new memory space using the routine on page 92, paragraph 12.3.1, of the handbook.

This additional memory space is ideal for routines such as Renumber, Pack etc . Additional advantages are: the space is free from corruption even during high resolution floating point work, unlike the normally proposed locations for such routines; and only one's own routines can write to this space, so preventing inadvertent erasure.



Final wiring with four new chips



Solder all pins except number 8.
This pin on the top 2114 is bent outwards.

Your Letters

Problems, queries and comments should be addressed to Letters, Acorn User, 53 Bedford Square, London WC1B 3DZ. The Editor regrets that it may not be possible to reply to every letter individually.

No business

Sir,
I started purchasing *Acorn User* through a newsagent, but stopped when it appeared to concentrate on games and graphics.

The initial articles on the BBC for business looked promising, but petered out. If you could assure me that this series could be developed, I would certainly order the next 12 issues.

Donald Forbes

These articles mentioned were featured in September's issue. As they explained, serious business use depends on the availability of disc drives which, so far, are few and far between.

A series on printing and printers has been commissioned and disc drives are being tested. Word processing packages and business software are also under scrutiny.

As the hardware and software becomes available these promising articles will be followed up.

Cassette choice

Sir,
I have cancelled my order for a cassette recorder to use with the BBC microcomputer because of non-delivery since ordering in February.

I should be pleased if you would recommend a suitable recorder.

F. G. Hilyer

Several readers have recommended a Boots cassette which costs about £25. Acorn supply a Ferguson recorder with their schools package. Although any recorder should work.

The best technique is to save programs at full tone and volume and load them back in at a medium setting. Loading commercial software should be no problem, but the cassette volume and tone may have to be altered.

Stereo recorders can be used, but machines with an automatic recording level can be troublesome.

Could do better

Sir,
Having read the first four issues of *Acorn User*, perhaps now is the time to accept your invitation to offer comment.

Congratulations to all concerned on a good beginning to a potentially successful magazine. It should prove essential reading for the thousands who, like me, are using the BBC micro as an introduction to computing.

One of the most pleasing aspects of the launch is your apparent willingness to make adjustments as you proceed. Some of the more obvious effects of early misjudgements having been progressively eliminated. It is to be hoped that the trend will continue.

Special thanks and congratulations are offered to Mr Telford who, despite the use of convoluted grammar, has provided such informative and helpful articles under the general heading of 'Hints and Tips'. Please continue to allow him plenty of space so that he might have room to explore in more depth some of the more complex aspects of computer programming. And please, try not to block out that which he has written with unnecessary artwork (p.34 October).

Speaking as a mature (perhaps over-ripe!) person, entranced by a hobby for which I am developing an insatiable thirst for knowledge, I for one shall assess the ultimate worth of the magazine more on the quantity and the quality of the relevant information it provides than by displays of unbridled enthusiasm on the part of the art department (much as I accept the pulling power of an attractive cover).

The switch to clearly printed listings is greatly appreciated (certainly by my tired old eyes). Any move towards an increase in the amount of useful information achieved by better use of space would, one feels, be welcomed, perhaps by the majority of readers.

E. Asquith

Many thanks for the comments which are always noted – and often acted upon.

There is a compromise to be made between artwork, density of text and size of illustration. Readers like yourself prefer large program listings – but these take up a lot of space. Similarly with diagrams. As for artwork, this is justified to make articles visually interesting, especially for younger readers, and to give information.

We hope the balance is right, and certainly *Acorn User* **makes far less use of the space-gobbling illustrations common in many other magazines.**

Graphics Update

Sir,
Could you tell me how it is possible to update a graphic display via the RS243 port or to send a message onto the screen without affecting the display, on a BBC model B computer?

It would be interesting to see an article on how to use the RS243 port to its full, ie talking to another computer, used as an intelligent terminal.

Are there any books giving details of the hardware of the BBC computer?

B. Mayne

It is only possible to update the graphic display in this way if you have the 1.2 version of the BBC operating system. When *FX2,1 is called, all input comes from the RS423 port.

Methods of using the Beeb as an intelligent terminal are under development. *Acorn User* **is keeping its ears to the ground on this one.**

On your final point, there are no books detailing the BBC machine's hardware available at present to our knowledge. If you have any specific requests, we will be pleased to try to answer them.

Mistaken variations

Sir, I am writing to draw your attention to the fact that the program printed for 'Text variations' on page 56 of the November issue, contains errors which prevent its operation.

The errors I have noticed are:

130 The number of the line is missing.
160 There are two lines 160. The first should be deleted.
210 W% should be Q%.
310 B should be B%.

With these alterations the program will run, although modifications are desirable to make it easier to enter charcters of various colours.

I am sure other readers will have drawn your attention to these errors which should be corrected because the program may be very useful in some circumstances.

George Foot

Yet again, our customer is right. To set the record straight, here is the correct version of lines 120–160:

```
120 Y% = H% - 991
130 FOR G% =1 TO 16 STEP 2
140 X% = G%:M% = 2
150 IF r=1 OR r=4 THEN
    X%=G%*2:M%=4
160 IF r=2 OR r=5 THEN
    X%=G%*4:M%=8
```

This listing was, unfortunately, not printed out from cassette, and the errors crept through when corrections were made to the typeset program.

Expensive?

Sir,
I did not think much of *Acorn User*, judging by the first two copies, but I have changed my mind since I bought the November issue.

I congratulate you on your program listings. They are excellent, very clear and a vast improvement on other publishers listings. The articles are also good and there is much of interest. I do however, think £1 is a high price, 80p would be nearer the mark.

You are no doubt aware of the errors in the Trek program, and it could be improved. All in all, though, a good program from which I have learned quite a bit.

I have yet to try the others.

Please continue your superb program listings.

Paul Whyte

The cover price may seem expensive, but good-quality printing and decent listings are time consuming and expensive. Also, we are confident *Acorn User* compares favourably in value for money terms with any computer magazine. Take your November issue and try it! We also pay very well for articles.

Memory, batteries and cartridges

Sir,
I have had my model B for about four months and the following points have surfaced:

i) Is it possible to use a battery to support the memory of the computer only, so a program can be retained while the mains is switched off?

ii) Can the 'shift lock' and 'caps lock' be controlled by a program?

iii) How does one flush only the keyboard buffer with the 0.1 operating system?

iv) Can the computer control household appliances, such as a hi-fi cassette deck with remote control?

v) When will cartridges be available to plug into the hole on the left of the keyboard and what will they do?

vi) Is it possible to alter the keys which control Acornsoft's *Defender (Planetoid)* game to avoid wear on the return-fire key?

vii) What's all this about an OS 1.2EPROM, and will it improve my OS 0.1?

viii) Why do I only get 28,158 bytes when I type PRINT HIMEM-TOP immediately after switching on, and not 32k of accessible memory left?

ix) Is it normal that to save a screen on tape around 50 blocks are recorded?

Ian Turner

The answer to the first two questions is, I'm afraid, no. And for the third. The fourth question raises two issues. It is fairly simple to use the cassette motor control relay to switch a larger relay to control the mains supply. However, controlling mains equipment by sending signals along the household mains wiring is more complicated. This requires transmitter units which plug into the micro and receiver units linked to equipment in other parts of the house. These are becoming commercially available.

On to point five – cartridges. These will be used to store programs (typically games) so they can be automatically run by the computer, and also extra speech vocabulary for the voice ROM.

The hole has been covered over on later machines as cartridges will not be available until after this year.

Acorn don't see wear on keys as a problem (even with *Defender*), because the machines are made to take this sort of punishment.

Acornsoft staff have notched up over 410,00 on *Defender*, 300,000 on *Snapper*, 110,000 on *Monsters* and 43,000 on *Meteors*. And yet they know of only one keyboard which gave problems – and that was on a pre-production model where the space bar became 'unreliable' after nine months! So if your keyboard wears out let Acorn know.

Analogue input can be used with games – essentially this is a joystick. Acornsoft packages are being adapted to take joysticks, and all future games will be able to do so. However, these are nowhere near as good as arcade versions and all the high scores given above were achieved using the keyboard. The message from games fanatics is that keyboards are best!

The BBC operating system is being continually developed by Acorn, with the later versions being downward compatible, ie if it works on the 1.2 system it will work on the 0.1 system. However, later operating systems contain new features. For example the 1.2 version will have 'fill area' and other facilities.

Putting operating systems into expensive EPROMs is only a temporary technique. Future operating systems will be mass-produced in ROM. Acorn will exchange EPROMs for ROMs through dealers (but check they've got them first!).

You do not have the full 32k available because some RAM is needed by the machine operating system – for example, to remember the 'time'. On disc machines this command gives the answer 25,342, showing that the disc system reserves over 2.5k of memory. Similarly the Econet interface will use up memory also. (To over-ride this, see Joe Telford's article in this issue.)

On your final point, yes. One block is 256 bytes, hence (quick bit of maths) 50 blocks equals 12,800 bytes. A mode 1 screen uses 20,479 bytes. The command *SAVE SCREEN 3000 7FFF will take 80 blocks to save and just manages – by one byte – the mode 1 screen.

The dealer list is on page 63

Ferocious micros

Sir,
Everyone says how 'user friendly ' the BBC micro is and I have certainly been most impressed by it. However, there are certain circumstances when even the most friendly machine becomes like a Victorian governess – when it doesn't like a program.

I use the fix you gave in September's *Acorn User* to get round the saving defect in the 0.1 operating system, but occasionally I save a half completed program and come back to it only to be faced with a screen full of error messages.

Please can you publish the machine code equivalent of 'I don't care how many missing headers, inadequate data etc you find, just don't ask

questions and put everything that can be interpreted as a character into the memory and I will list it and sort it out later!'

Roger Kemp

You asked for it, so here it is – *OPT 2,0. This command ensures that the Beeb ignores all errors found during cassette loading, although messages will continue to be given.

Details of using *OPT are given in the encyclopaedic *User Guide* **(page 398).**

Tricky Trek

Sir,

I own a BBC model B micro and receive *Acorn User* on subscription.

After receiving the November issue three days after I'd seen it in the local shops, I entered the program *Trek III* and was disappointed to find a

● The program doesn't check for a valid difficulty. Insert the follow lines:

```
35 REPEAT
45 UNTIL D>4 AND D<41
```

The Enterprise should be preceded by YC$ in lines 1190,1200 and 1220; the omission of this causes the ship to appear blue and causes difficulty with aiming phasers, ie you have to be positioned one column to the left of them and 'docking' is similarly affected. This can be cured by using enterprise $=YC$ + appropriate character in the lines mentioned above.

There is no limit to the number of photons used and the armaments computer can display negative numbers. This can be cured by inserting the following line which will cause a return to SRS if attempting to fire a non-existent photon torpedo.

```
805 IF photon < 1 THEN 160
```

I enjoy reading your magazine and find some of the articles very helpful. Would it be possible to show how to synchronise pitch and amplitude envelopes? I hope that in future I will receive *Acorn User* before, not after, it appears on the local bookshelves as I am paying more.

Andrew Mooney

Several readers suggested these changes in *Trek III*, **and we are grateful to them for drawing our attention to the problems, although the program will run if input correctly.**

The additions suggested by Tim Heaton are:

1 ON ERROR *FX4,0:END
45 IF D<5 OR D>40 GOTO 1
790 ON B GOTO 160,795,820,480
795 IF photon = 0 GOTO 160
880 PRINTTAB (32,15) CHR$130; shield "__ __ __"
1190 *for* "[" *insert* YC$ + "["
1200 *for* "]" *insert* YC$ + "]"
1220 *for* "^" *insert* YC$ + "^"
1845 IF shield <10 shield =0
1850 for shield insert shield; "— — —"

On the subject of subscriptions, delays in delivery have been caused by the late clearance of credit cards (in some cases this has taken six

weeks, even though the card is debited immediately), the time taken to set up subscriptions and, on the December issue, by the loss of address labels (oops!).

However, all these problems should have been sorted out for this issue. So if it's late let us know!

● To other readers who had difficulties inputting programs. First check the difference beween 1 and l. This can be seen in line 70, as 1 has a longer bar at the top. Ensure lower case characters are printed lower case, not capitals.

Also note that in mode 7 (when the Beeb is first switched on) certain symbols look different on the screen. This is explained in the *User Guide* (p.18) and will not affect the program.

It is the policy of *Acorn User* to print listings direct from cassette as often as possible to avoid errors. In *Trek III* it was an early version of the game on the same cassette which was printed! We have also arranged a new typeface to avoid the 1 and l problem.

Video patterns

Sir,
I have connected the video output of my model B BBC computer to the input socket of my video recorder and have been able to tape the computer's output, but the screen is completely marred by 'moire' patterns. The output of the recorder can only be fed into the domestic television set through the aerial socket, as my television has no direct input socket for a video recorder, though it does have a separate button for VCR input, which is used.

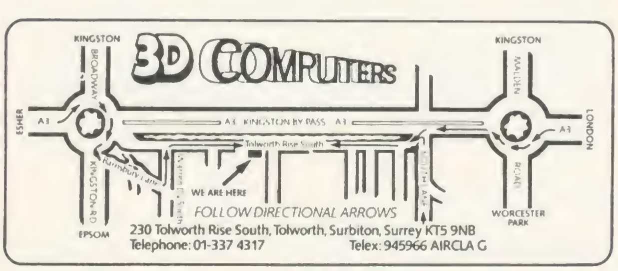
The only practical use I can think of for this set up, which I tried out of curiosity only, is to insert titles onto videotapes, but I should still like to be able to do it. Is the pattern effect due to closeness of frequencies used by the VCR and the computer? If it might be cured by the insertion of a choke in the lead could you suggest some possible values?

A. Collins

The problem here is that the colour subcarrier is not phase-locked to the line and field oscillators on the BBC micro – a problem common to most micros. The moire patterning can be reduced by altering the subcarrier frequency. This is controlled by VC1 which is located in the north-east corner of the Beeb's printed circuit board.

This technique can, as you say, be used to record titles onto videotape, but not to overlay titles on pictures. Only if the computer is 'slaved' to the picture can this be done.

BBC television studios have developed this modification and a board will probably be produced in late 1983 to enable the BBC micro to be used as a synchronous source. However, this is intended for professional studios rather than home video users.


3D COMPUTERS
KINGSTON
BROADWAY
KINGSTON
MALDEN
ESHER
LONDON
A3 KINGSTON BY PASS A3
Tolworth Rise South
WE ARE HERE
FOLLOW DIRECTIONAL ARROWS
KINGSTON RD
EPSOM
WORCESTER PARK
230 Tolworth Rise South, Tolworth, Surbiton, Surrey KT5 9NB
Telephone: 01-337 4317
Telex: 945966 AIRCLA G

Official Acorn dealers

Acorn dealers stock and service the Atom computer, Acorn systems and Acornsoft software. Many also offer service facilities for the BBC micro and these are marked with a ● symbol. The Acorn dealer not only sells computers and peripherals but provides vital customer support. Most have recently attended technical seminars in Cambridge to ensure that they deal effectively with customers' enquiries. In addition, Acorn supply dealers with specific test and diagnostic equipment to speed fault finding.

LONDON
Cavendish Sales E1
01-247 3453
Canonbury Radio
01-226 9392
● Centre of Sound
01-727 0511
● Group 70 E18
01-505 7724
● Off Records SW11
01-223 7730
Paul Electrical SW20
01-542 6546
PAJ Rental Ltd NW4
01 368 0572
REW West End
Video Centre WC2
01-240 3386/7
RA Bailey Ltd SE26
01 778 8101
● Technomatic Ltd
NW10
01-452 1500
Technomatic Ltd W2
01-723 0233
The Video Palace W8
01-937 8587

BIRMINGHAM
Quality Radio &
Television Co B14
021 444 2889

AVON
● Microstyle
Bath
0225-334659

BEDFORDSHIRE
● Broadway Electronics
Ltd
Bedford
0234-213639

BERKSHIRE
Microstyle
Berkshire
0635 41929
● Windsor Computer
Centre
Windsor
07535 58077

BUCKINGHAMSHIRE
A L Wheeler Ltd
Great Missenden
024-06 2560
Hi-Vu Electronics
Wolverton
0908 312808

CAMBRIDGESHIRE
● Arden Personal
Computers
Peterborough
0733-47767
● Cambridge Computer
Store
Cambridge
0223-65334
● Control Universal
Cambridge
0223-358757

CHESHIRE
● Northern Computers
Warrington
0928-35110

CLEVELAND
● Customised
Electronics Ltd
Middlesbrough
0642-247727

CORNWALL
● Brewer & Bunney
Camborne
0209 712681
Microtest Ltd
Bodmin
0208 3171

DERBYSHIRE
● Datron Micro Centre
Derby
0332-380085
● First Byte Computer
Systems Ltd
Derby
96 365280

DEVON
● Bits & Bytes
Ilfracombe
0271-62801
Devon Computers
Paignton
0803-526303
J A D Integrated
Services
Plymouth
0752 62616

DORSET
Classic Office
Equipment
Bournemouth
0202 28666
Landsowne Computer
Centre
Bournemouth
0202 23776

ESSEX
● Arden Business
Systems
Colchester
020-637 368
● Computers for All
Romford
0708-60725

GLOUCESTERSHIRE
Computer Shack Ltd
Cheltenham
0242 584343
Independent Computer
Consultants
Tewkesbury
0684 298033

HAMPSHIRE
● Electronequip
Porchester
0705-325 354
Basingstoke Computer
Centre
Basingstoke
0256 52203
● R D S Electrical
Portsmouth
0705-812478
● R M K Electronics Ltd
New Milton
0425-616110

HEREFORDSHIRE
Kempson's
Hereford
0432 3480

HERTFORDSHIRE
● Compshop Ltd
New Barnet
01 441 2922
● Computer Plus
Watford
0923 33927
H F Sheffield Ltd
Abots Langley
092 77 63184
Intelligent Artifacts Ltd
Royston
0233 207689
● Q-Tek Systems Ltd
Stevenage
0438 65385

HUMBERSIDE
● Computer Facilities
Scunthorpe
0724 63167
● Vixon Computer
Systems
Cleethorpes
0472 58561

KENT
Kent Computers
Herne Bay
02273-68900
Kent Micro Computers
Maidstone
0622 52784
● M D Wright Data
Services
Canterbury
0227-69090

LANCASHIRE
Almaine Co
Colne
0282 863520
J Lambert (Radio) Ltd
Burnley
0282 71459
Merit Computers Ltd
Wigan
0942-495821
Microrose Ltd
Bentham
0468 62180
● NSC Computing
Shops Ltd
Manchester
061-832 2269

LEICESTERSHIRE
● D A Computers
Leicester
0533 549407
Percy Lord & Son Ltd
Wigston
0533 785033
Pratt Bros (Leicester)
Ltd
Blaby
0533 773020
● R H Murcott Ltd
Loughborough
0509 214444

LINCOLNSHIRE
● Oakleaf Computers
109 Dudley Street
Grantham

**LIVERPOOL/
MERSEYSIDE**
BEC Computer World
66 Lime St Liverpool
● Data Exchange Ltd
Birkenhead
051-647 4213
Tisdails Newton-le-
Willows
09252 5577

NORTHAMPTONSHIRE
● MA Electrical
Installations
Irthlingborough
0933 650133

NORTHUMBERLAND
Tom Newton & Sons
Wooler
066 82221

NOTTINGHAMSHIRE
● Leasalink Viewdata
Nottingham
0602 396976
H N & L Fisher
(Huthwaite) Ltd
Sutton in Ashfield
0623 53435
● Mansfield Computers
& Electronics
79 Ratcliffe Gate
Mansfield
P R Hartley
Ruddington
0602 213492
S P Electronics
Hucknall
0602 632467

MIDDLESEX
● Microage Electronics
Edgware
01-959 7119
● Twickenham
Computer Centre
Twickenham
01-891 1612

NORFOLK
● Anglia Computer
Store
Norwich
0603 29651

SHROPSHIRE
Com-tel
101 High St Newport
● Jentech Services Ltd
Bridgenorth
07462 5287

SOMERSET
● Somerset Business
Computers
Taunton
0823-52149
● The Computer Room
Yeovil
0935-20268

SUFFOLK
CE Matthews & Co
Ipswich
0473 215666
● Micro Management
Ipswich
0473 59181
Midwich Computer Co
Ltd
Rickinghall
0379 898751
S Emery & Co
Bungay
0271 62503

SURREY
Croydon Computer
Centre
01-689 1280
● JS Simnett
Computers Ltd
Kingston upon Thames
01-546 3793
● PJ Equipments Ltd
Guildford
0483-504801
● 3D Computers
Surbiton
01-337 4317

SUSSEX
● Castle Electronics
Hastings
0424-437875
● Datex Applied Micros
Worthing
0903-211011
Gamer
Brighton
0273-698424
● Microcentre
Bognor Regis
0243 827779

TYNE AND WEAR
● H C C S
Gateshead
0632-821924
Newcastle Computer
Services
Newcastle-upon-Tyne
0632-761168

WARWICKSHIRE
Carvell
Rugby
0788 65275

YORKSHIRE
Arthur Yates Ltd
Ripon
0765-2737
● Customised
Electronics Ltd
Leeds
0532-792332
● Datron Computers
& Supplies
Sheffield S10
0742-755105
● Datron Micro Centre
Sheffield S7
0742-585490
● Eltec Services Ltd
Bradford
0274-491371
Green's Telecom
Barnsley
0226-5031
Micro Power
Leeds LS7
0532-683186
Pennine Computers Co
Ltd
Halifax
0422 41719
● Superior Systems Ltd
Sheffield 1
0742-755005

WEST MIDLANDS
A E Chapman & Co
Reddal Hill Rd
Cradley Heath
D F Gibbs Ltd
Coventry
0203 87432
Richard Morris
(Electrical) Ltd
Warley
021-429 1161
● Walters Computer
Systems Ltd
Stourbridge
03843-70811/2/3

WALES
● Bucon Ltd
138 Walter Rd
Swansea SA1 5RQ
● Cardiff
Microcomputers
Cardiff
0222 373072
Cardigan Electronics
Cardigan
0239 614483
KB Computers
Llangollen
0691 72 651
NE Wales
Microprocessor Centre
Flint
0244 816236
SIR Computers
Cardiff
0222 616065

SCOTLAND
● Esco Computing
Glasgow
041-4275497
● Highland
Microcomputers
Inverness
0463-221544
Ian Hutchison (Leven)
Ltd
Leven
0333-23870
J H Donald Ltd
6 Riccarton Rd Hurlford
McQueen Systems Ltd
Galashiels
0896 4866
Personal Computers
West Coast
Ayr
0292 285082
Service Centre
Greenock
0475 20228
Silicon Centre
Edinburgh
031 557 4546
W M Coupar
Blairgowrie & Dundee
0382 2436/7
Gate Microsystems
Dundee
0382 28194

NORTHERN IRELAND
● C E M Microcomputer
Services
Belfast
0232-44111/43564

CHANNEL ISLANDS
● Channel Data
Processing (Jersey)
St Helier
0534 24333/72897

ISLE OF MAN
● Typestyle Ltd
Onchan
0624-25890/24650

ISLE OF WIGHT
Excell
Bembridge
098 387 2578

PROGRAMPOWERMICROPOWER

GREAT BBC PROGRAMS FROM BRITAIN'S LEADING SOFTWARE HOUSE

SWOOP


SWOOP


ALIEN DESTROYERS


CHESS

SWOOP (B) £6.95 – the NEW GALAXIANS IT'S HERE AT LAST!! Galaxian-style, machine code arcade game. THIRTY screaming, homing, bomb-dropping, explosive egg-laying BIRDMEN, swooping down in ones and two's to destroy your laser bases. The exploding eggs feature makes a normally difficult game into a challenge 'par excellence.' Each new screen means increased difficulty. Bonus bases, score display, high-score and rankings are, of course included. YOUR WAITING IS OVER!!

ALIEN DESTROYERS (B) £6.95
Sensational, high speed 'INVADERS' program with an abundance of features. Brilliant use of sound and graphics. 48 strong Alien Fleet of three different types plus Mothership scoring mystery bonus. Choice of six alien speeds and three bomb speeds. Vertical, angled and exploding missiles. Options to replace defences and suppress new fleet advances. Bonus bases awarded each new sheet. Scoring according to overall difficulty level. Ongoing display of score and hi-score. End of game rankings of top five scores.

This program has many unique extras e.g. 'battle analysis' showing the number of each alien type shot down, how many motherships destroyed, the number of sheets cleared, the shots fired, the percentage of hits made and the number of bases lost.

CHESS (B) £6.95
Our excellent machine code program – now with superb MODE 1, colour graphics. Six skill levels, play black or white, illegal moves rejected, 'en passant', castling, take-back of moves, and display of player's cumulative move-time. Options include Blitz Chess where you must move in 10 seconds, set-up of positions for analysis, replay of a game just played and saving of part completed games on tape. On loading, a 1972 Spassky/Fischer game can be replayed.
NOTE: Model A Version still available at only £4.95.
If you wish to upgrade your Model A version please return your tape, together with £2.50 plus V.A.T.

Other B.B.C. programs available:
Galactic Commander (B) £6.95/ Laser Command (B) £6.95/ Adventure £6.95/ Cowboy Shoot-Out (B) £5.95/ Filer £8.95/Micro Budget £6.95/ World Geography (B) £5.95.
Timetrek (B) £6.95/Spacemaze (B) £5.95/Martians (B) £5.95/Astro Navigator (B) £4.95/Star Trek £4.95/Maze Invaders (B) £4.95/Footer (B) £6.95/Munchyman £5.95/Seek £5.95/Eldorado Gold (B) £5.95/Cat & Mouse £4.95/ Mastermind £3.95/Reversi 1 £4.95/Reversi 2 (B) £4.95/ Roulette (B) £4.95/Gomoku £3.95/Zombies £3.95/ Dissassembler £5.95/Constellation (B) £5.95/ Where? (B) £5.95/Junior Maths Pack (B) £5.95.

WE Guarantee THAT ALL OUR ADVERTISED PROGRAMS HAVE BEEN COMPLETED AND ARE READILY AVAILABLE

WRITTEN ANY PROGRAMS! WE PAY 20% ROYALTIES FOR DRAGON, SPECTRUM BBC, ATOM PROGRAMS

WE ARE AUTHORISED DEALERS FOR ACORN ATOM, BBC MICRO & DRAGON 32

SPECIAL OFFER
Deduct £1 per cassette when ordering two or more.

MICRO POWER LTD.
8/8a REGENT STREET,
CHAPEL ALLERTON,
LEEDS LS7 4PE.
Tel: (0532) 683186

Please add 55p order P & P + VAT at 15%

Please Note:
All programs are now available at all good dealers or direct from MICRO POWER LTD.

PROGRAMPOWERMICROPOWER